

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації та управління*

УДК 004

«До захисту допущено»

**В.о. завідувача кафедри**

О.А. Павлов

(підпис)

(ініціали, прізвище)

“ ” \_\_\_\_\_ 2019 р.

**Дипломний проект**  
**на здобуття ступеня бакалавра**

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: Підсистема підтримки діяльності малого та  
середнього бізнесу у складі автоматизованої банківської системи  
“БАРС”

**Виконала:**

студент 4 курсу, групи ІС-51

Сборик Антон Юрійович

(прізвище, ім'я, по батькові)

(підпис)

**Керівник**

доц., к.т.н., доц. Телишева Т.О

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

**Консультант з  
графічної  
документації**

ст. викл. Москаленко Н.В

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

**Рецензент**

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050101 «Комп'ютерні науки»

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

О.А. Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Сборику Антону Юрійовичу  
(прізвище, ім'я, по батькові)

**1. Тема проекту** Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”

керівник проекту Тєлишева Тамара Олексіївна, доц., к.т.н., доц.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від “23”квітня 2019 р. №1181-с

**2. Строк подання студентом проекту** “03”червня 2019 року

**3. Вихідні дані до проекту**

Технічне завдання

**4. Зміст розрахунково-пояснювальної записки**

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

## 5. Перелік графічного матеріалу

1. Схема структурна діяльності

2. Схема структурна варіантів використання

3. Схема бази даних

4. Схема структурна послідовності

5. Схема структурна класів програмного забезпечення

6. Схема структурна АС

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «15» лютого 2019 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	15.04.2019	
2.	Аналіз існуючих методів розв'язання задачі	20.04.2019	
3.	Постановка та формалізація задачі	21.04.2019	
4.	Розробка інформаційного забезпечення	25.04.2019	
5.	Алгоритмізація задачі	01.05.2019	
6.	Обґрунтування використовуваних технічних засобів	03.05.2019	
7.	Розробка програмного забезпечення	05.05.2019	
8.	Налагодження програми	13.05.2019	
9.	Виконання графічних документів	15.05.2019	
10.	Оформлення пояснювальної записки	18.05.2019	
11.	Подання ДП на попередній захист	22.05.2019	
12.	Подання ДП на основний захист	03.06.2019	
13.	Подання ДП рецензенту	06.06.2019	

Студент

\_\_\_\_\_ Сборик А.Ю.  
(підпис)

Керівник проекту

\_\_\_\_\_ Телишева Т.О.  
(підпис)

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проект	1	
			<b><u>Документи текстові</u></b>		
2	A4	ДП ІС-5124.1181-с.ПЗ	Пояснювальна записка	92	
3	A4	ДП ІС-5124.1181-с.ТЗ	Технічне завдання	12	
4	A4	ДП ІС-5124.1181-с.ПМВ	Програма та методика випробувань	17	
			<b><u>Матеріали графічні</u></b>		
5	A4	ДП ІС-5124.1181-с.ССД	Схема структурна діяльності	1	
6	A3	ДП ІС-5124.1181-с.ССВ	Схема структурна варіантів використання	1	
7	A4	ДП ІС-5124.1181-с.СБД	Схема бази даних	1	
8	A3	ДП ІС-5124.1181-с.ССП	Схема структурна послідовності	1	
9	A3	ДП ІС-5124.1181-с.ССК	Схема структурна класів програмного забезпечення	1	
10	A3	ДП ІС-5124.1181-с.САС	Схема структурна АС	1	

					<b>ДП ІС-5124.1181-с.ВДП</b>		
<b>Зм.</b>	<b>Арк.</b>	<b>ПІБ</b>	<b>Підп.</b>	<b>Дата</b>			
Розробн.		Сборик А.Ю.			<b>Відомість дипломного проекту</b>	<b>Лист</b>	<b>Листів</b>
Керівн.		Телишева Т.О.				<b>1</b>	<b>1</b>
Консульт						НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ кафедра АСОІУ гр. ІС-51	
Н/контр.		Москаленко Н.В.					
З.каф.		Павлов О.А.					

# **Пояснювальна записка до дипломного проекту**

на тему: Підсистема підтримки діяльності малого та середнього  
бізнесу у складі автоматизованої банківської системи “БАРС”

Київ – 2019 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з шести розділів, містить 29 рисунків, 19 таблиць, 1 додатків, 15 бібліотечних найменувань.

Основними цілями розробки є:

- зменшення часу на відкриття депозитів представникам малого бізнесу, середнього бізнесу та працівниками банку за рахунок автоматизації процесу формування документів на основі введених даних;
- пониження ймовірності помилок при оформленні документів працівниками банку за рахунок валідації форм введення даних та додаткової перевірки даних збережених до бази даних працівниками бек офісу.

В дипломному проекті удосконалено підсистему роботи з депозитними договорами клієнтів банку – юридичних осіб автоматизованої банківської системи “БАРС”.

В результаті впровадження розроблених рішень зменшилась ймовірність помилок при оформленні депозитів та скоротився час оформлення депозитів для малого та середнього бізнесу.

Підсистема “Депозитний модуль” у складі АБС “БАРС” планується до використання.

**ДЕПОЗИТ, АВТОМАТИЗОВАНА БАНКІВСЬКА СИСТЕМА, КАПІТАЛІЗАЦІЯ, БІЗНЕС-ПРОЦЕС, БАЗА ДАНИХ, МАЛИЙ ТА СЕРЕДНІЙ БІЗНЕС**

					ДП ІС-5124.1181-с.ПЗ							
Зм.	Арк.	Прізвище	Підпис	Дата								
Розроб.		Сборик А.Ю.			Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”			Літ.	Арк.	Аркушів		
Перевірив.		Тєлшиєва Т.О.								2	92	
								НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ кафедра АСОІУ гр. ІС-51				
Н. кон.		Москаленко Н.В.										
Затв.		Павлов О.А.										

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of six sections, containing 29 drawings, 19 tables, 1 application, 15 sources.

The main objectives of the development of the deposit module are:

- reducing the time for opening deposits to representatives of small business, medium-scale businesses and bank employees by means of automation of the process of forming documents based on the data entered;
- reduction of the probability of mistakes in the execution of documents by bank employees by validating the forms of data entry and additional verification of data stored in the database by the employees of the back office.

In the diploma project the subsystem of work with deposit agreements of bank clients – legal entities of the automated banking system "BARS" – has been improved.

As a result of the implementation of the developed solutions, the probability of errors in the process of making deposits reduced and the time for the deposit of small and medium-sized businesses decreased.

Subsystem "Deposit module" in the ABS "BARS" is planned to be used.

DEPOSIT, AUTOMATED BANKING SYSTEM, CAPITALIZATION,  
DATABASE, BUSINESS-PROCESS, SMALL AND MEDIUM-SCALE  
BUSINESS

					ДП ІС-5124.1181-с.ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

<b>СПИСОК СКОРОЧЕНЬ .....</b>	<b>5</b>
<b>1. ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>6</b>
1.1. Опис предметного середовища .....	6
1.1.1. Опис процесу діяльності при оформленні депозиту .....	11
1.1.2. Опис функціональної моделі оформлення депозиту .....	13
1.2. Огляд наявних аналогів.....	14
1.3. Концепція депозитного модуля.....	15
1.3.1. Опис результатів вивчення об'єкту автоматизації.....	15
1.3.2. Опис та оцінка переваг і недоліків розроблених альтернативних варіантів концепції створення модуля .....	16
1.3.3. Порівняльний аналіз вимог користувача до підсистеми і варіантів концепції підсистеми на предмет задоволення вимог користувача .....	16
1.3.4. Очікувані результати та ефективність реалізації обраного варіанту концепції .....	17
1.3.5. Орієнтовний план реалізації обраного варіанту концепції підсистеми .....	17
1.3.6. Необхідні витрати ресурсів на розробку, введення в дію та забезпечення функціонування .....	18
1.3.7. Вимоги, що гарантують якість підсистеми.....	18
1.3.8. Умови приймання підсистеми .....	18
1.4. Постановка задачі .....	19
1.4.1. Призначення розробки .....	19
1.4.2. Цілі та задачі розробки .....	19
<b>Висновки до розділу .....</b>	<b>20</b>
<b>2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>21</b>
2.1. Вхідні дані підсистеми ВД .....	21
2.2. Вихідні дані підсистеми ВД .....	22
2.3. Опис структури бази даних .....	26
<b>Висновки до розділу.....</b>	<b>29</b>
<b>3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>30</b>
3.1. Змістовна постановка задачі розрахунку результатів депозиту .....	30



3.2.	Математична постановка задачі розрахунку результатів депозиту .....	31
3.2.1.	Задача знаходження суми коштів по відсотковій ставці .....	31
3.3.	Обґрунтування методу розв'язання.....	31
3.3.1.	Задача знаходження суми коштів по відсотковій ставці .....	31
3.4.	Опис методів розв'язання .....	31
3.4.1.	Задача знаходження суми коштів по відсотковій ставці .....	31
	Висновки до розділу.....	33
<b>4.</b>	<b>ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>34</b>
4.1.	Засоби розробки.....	34
4.2.	Вимоги до технічного забезпечення .....	37
4.2.1.	Загальні вимоги .....	37
4.3.	Архітектура програмного забезпечення підсистеми “Депозитний модуль” .....	38
4.3.1.	Діаграма послідовності .....	38
4.3.2.	Діаграми класів .....	38
4.3.2.	Діаграма розгортання.....	40
4.3.3.	Специфікація функцій .....	40
	Висновки до розділу.....	44
<b>5.</b>	<b>ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>45</b>
5.1.	Керівництво користувача операторів фронт і бек офісу .....	45
5.2.	Випробування програмного продукту підсистеми “Депозитний модуль” .....	58
5.2.1.	Мета випробувань .....	58
5.2.2.	Загальні положення .....	58
5.2.3.	Результати випробувань .....	58
	Висновки до розділу.....	67
	<b>ЗАГАЛЬНІ ВИСНОВКИ .....</b>	<b>68</b>
	<b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>	<b>69</b>
	<b>ДОДАТОК А .....</b>	<b>71</b>

## СПИСОК СКОРОЧЕНЬ

АБС	Автоматизована банківська система
АК	Анкета клієнта
БД	База даних
ДД	Депозитний договір
ДМ	Депозитний модуль
ЄДРПОУ	Єдиний державний реєстр підприємств та організацій України
ЗВД	Заява на відкриття договору
ІПН	Індивідуальний податковий номер
НБУ	Національний Банк України
ADO	(ActiveX Data Objects) Об'єкти даних ActiveX
DOM	(Document Object Model) Об'єктна модель документа
IDE	(Integrated Development Environment) Інтегрована середовище розробки
MVC	(Model-View-Controller) Модель-Представлення-Контроллер
SQL	(Structured Query Language) Мова структурованих запитів

## 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1. Опис предметного середовища

Підприємствами, що класифікуються, як підприємства малого бізнесу вважають підприємства, у яких дані у фінансовій звітності за минулий рік до звітного задовольняються хоча б двом із критеріїв:

- балансова вартість активів – до 4 мільйонів євро;
- чистий дохід від реалізації продукції (товарів, робіт, послуг) – до 8 мільйонів євро;
- середня кількість працівників – до 50 осіб[1].

Підприємствами, що класифікуються, як підприємства середнього бізнесу вважають підприємства, у яких дані у фінансовій звітності за минулий рік до звітного задовольняються хоча б двом із критеріїв:

- балансова вартість активів – до 20 мільйонів. євро;
- чистий дохід від реалізації продукції (товарів, робіт, послуг) – до 40 мільйонів. євро;
- середня кількість працівників – до 250 осіб[1].

Депозит – сума грошових та негрошових активів (цінних паперів, дорогоцінних матеріалів, акцій, облігацій), передана особою кредитній установі на встановлений строк з метою отримати прибуток у вигляді відсотків, що утворюються в ході фінансових операцій з вкладеними активами[2]. На рисунку 1.1 зображено різновиди депозитів для малого та середнього бізнесу.



Рис.1.1 Різновиди депозитів.

Змн.	Арк.	№ докум.	Підпис	Дата

У залежності від терміну на який здійснюється вклад активів, депозити можна розділити на дві категорії: депозити «до запитання» й строкові депозити.

Під депозитом «до запитання» розуміють ті активи, які вкладник може забрати з банку в будь-який час, шляхом переведення даних активів на рахунок іншої фізичної / юридичної особи, або вивести з рахунку шляхом зняття грошей в касі банку або скориставшись банкоматом в разі вкладення грошових активів. Залишки грошових активів на банківських рахунках клієнтів банків відносять до депозитів «до запитання» тому, що в деяких банках існують бонусні програми за якими нараховуються відсотки від залишку на рахунку клієнта банку.

Депозит «до запитання» є найкращим варіантом, якщо основною метою вкладення активів є їх зберігання, а не отримання прибутку оскільки активи, що знаходяться на банківському рахунку можуть зберігатися необмежений період, їх можна залишати по заповіту й відізнати в будь-який момент. Депозит «до запитання» – це безстроковий рахунок в банку, якому властиве автоматичне пролонгування. Характерною рисою депозиту «до запитання» є те, що він не обмежений по залишку і сумі внеску.

Основним недоліком депозиту «до запитання» являється відсоток нарахування мінімального прибутку, що становить 1,5%. Отже, цільовими користувачами даного виду депозитів є юридичні особи, які відкривають цей вид депозитного рахунку з ціллю зберігання активів у фінансовій установі, а не задля отримання прибутку.

**Строкові депозити** поділяють на три категорії:

- короткострокові (термін 1-3 місяці);
- середньострокові (термін 3-9 місяців);
- довгострокові (термін більше 9 місяців).

Особливістю строкових депозитів являється те, що вкладені активи можна отримати лише після завершення терміну зазначеного в договорі.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Вкладник отримує відсотки за те, що банк користується вкладеними активами, перетворюючи їх в кредити та надаючи їх іншим користувачам банку. Таким чином, кошти на депозиті приносять прибуток своєму власникові.

Методи нарахування відсотків по депозитному рахунку зображено на рисунку 1.2.



Рисунок 1.2 Методи нарахування відсотків.

Розглянемо більш детально деякі з методів нарахування відсотків по депозитному рахунку.

### З вказаним періодом

Від терміну на який вкладаються активи до депозитного рахунку залежать відсоткові ставки – чим довший час користування фінансовою установою активами вкладника, тим вищі відсоткові ставки. Знімати відсотки можна кожного місяця або капіталізувати їх на рахунку фінансової установи. Вкладник може перевести короткостроковий депозит після закінчення терміну дії на мінімальну ставку або депозит може автоматично продовжитися на новий період – даний розвиток процедур прописаний в договорі.

Варто відзначити, що користувач депозитного рахунку у будь-який момент може отримати необхідні йому активи, але в такому випадку вкладник втрачає відсотки. Деякі фінансові установи надають можливість клієнтам самим визначити термін, на який їм буде зручно розмістити активи. Такий термін має назву індивідуального терміну депозиту. Його перевагою є те, що споживач сам вибирає той час, коли йому будуть необхідні активи і отримує прибуток з депозитів.

Для розірвання депозитного договору раніше зазначеного часу або виплати частини вкладених активів за вимогою вкладника – юридичної особи, клієнт повинен сповістити про свої наміри банк, що надав депозитний рахунок, через подання заяви. Після отримання заяви про розірвання договору, банк реєструє їх за внутрішнім положенням, що встановлено в конкретному банку.

**Відсотки, що налічуються наприкінці депозитного терміну** сумуються до вкладу перед поверненням вкладнику. В такий самий спосіб нараховуються відсотки на грошові активи, що знаходяться на банківському рахунку. З певною періодичністю (щомісяця, щодня) в той самий час для певного періоду інформаційна система банку робить перевірку залишку коштів на банківському рахунку та нараховує відсоток, розраховуючи його за поточною відсотковою ставкою, що є нижчою за строкову депозитну ставку. В більшості випадків нарахування прибутку від відсотків відбувається щомісяця.

При використанні строкового депозиту також існує можливість отримання відсотків з певною періодичністю. При отриманні відсотків початковий вклад залишається сталим. Але в цей самий час вкладник отримує регулярний прибуток.

**При капіталізації депозитних відсотків** має місце збільшення початкового вкладу з певною періодичністю (щомісяця, щоквартала, щороку). Нарахування відсотків в наступному періоді відбувається на основі величини вкладу з попереднього періоду.

Також існують депозити, що надають можливість збільшити початковий вклад шляхом внесення деякої додаткової кількості грошових або негрошових активів. При збільшенні початкового вкладу відсоткова ставка буде збільшуватись пропорційно до вкладення.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

**Всі умови депозитних вкладів є прописаними в договорі.**

Найважливішими серед них є:

- відсоткова ставка за поточним депозитом;
- мінімальна і максимальна сума депозитного вкладу;
- терміни і порядок виплати або капіталізації відсотків;
- можливість додаткового поповнення поточного рахунку;
- умови дострокового закриття чи пролонгації.

Депозити також поділяють на депозити з можливістю поповнення та без можливості поповнення. Прикладом депозиту з можливістю поповнення є депозит «до запитання» (можна робити додаткові внески незалежно від часу з моменту вкладення). Строкові депозити за можливістю поповнення поділяють на:

- ощадні, які не передбачають можливості поповнення й зазвичай використовуються для накопичення активів;
- накопичувальні, що передбачають можливість поповнення на будь-яку суму. Але деякі банки встановлюють ліміти. Для накопичуваних строкових депозитів відсотки нараховуються на основі загальної суми вкладу.

Особливістю даних депозитів є менший відсоток порівняннл з ощадними депозитами. Причиною цього є те, що банк не ризикує з великою відсотковою ставкою тому, що не знає в момент укладення договору на створення депозитного рахунку майбутній розмір вкладу клієнта.

Також існують депозити з можливістю часткового виведення коштів, але в них визначається мінімальний розмір вкладів, що має залишитися на рахунку клієнта. Клієнт банку може знімати активи з рахунку, але він повинен повернути активи в такий спосіб, щоб сума активів на рахунку стала такою, як на момент відкриття депозитного рахунку. Особливістю даного типу депозитних договорів є нижчі відсоткові ставки.

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

### 1.1.1. Опис процесу діяльності при оформленні депозиту

Бізнес-процес оформлення депозиту наведено на рисунку 1.3 за допомогою діаграми IDEF0.

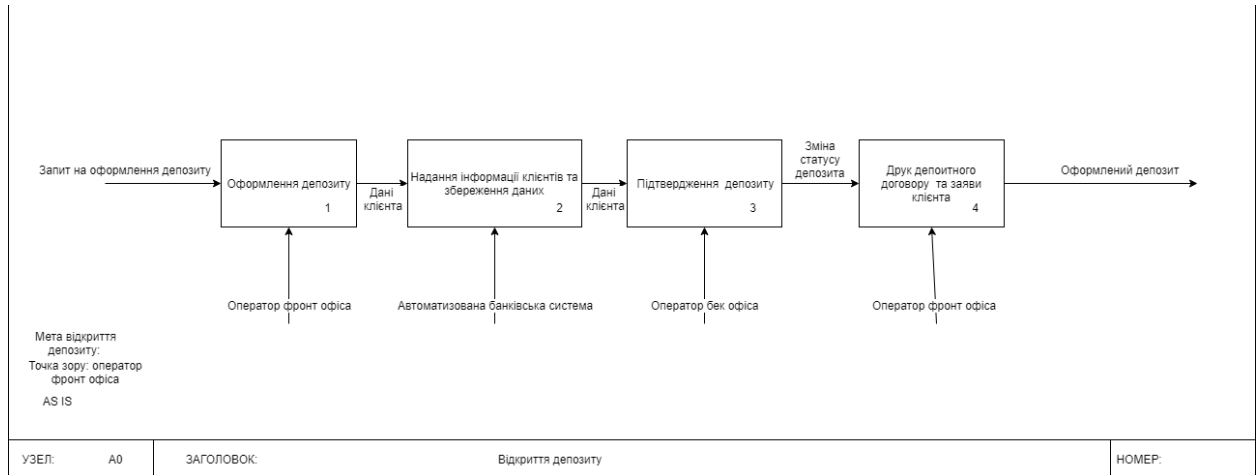


Рисунок 1.3 Діаграма бізнес процесу оформлення депозиту

Розглянемо процес відкриття депозиту зі сторони клієнтів банку. Основними етапами відкриття депозитного рахунку є:

- відвідування відділення банку;
- вибір типу депозитного продукту;
- заповнення заяви на відкриття депозитного рахунку та надання документів потрібних для оформлення депозиту;
- підписання договору клієнтом.

Після перевірки даних клієнта та підписання ним договору, договір також підписує оператор фронт офісу банку.

Юридичним особам необхідно надати документи паспорт та ідентифікаційний код та інші. Детальний перелік документів слід уточнювати в конкретному банку, куди юридична особа має намір зробити вклад.



Кожен договір, що укладається між банком та вкладником містить:

- предмет договору;
- обов'язки фінансової установи;
- права та обов'язки вкладника;
- спосіб вирішення спірних питань;
- можливість дострокового розірвання депозитного договору.

Договір визначає взаємовідносини між вкладником та фінансовою установою, до якої він зробив внесок.

Договір може містити спосіб виплати прибутку з відсоткової ставки, можливість внесення додаткових вкладів, можливість збільшення/ зменшення відсоткової ставки та інше.

Договір являється основним документом при вирішенні конфліктів у суді.

Після завершення терміну депозиту вкладник повинен закрити депозитний рахунок шляхом відвідування банку з договором на створення депозиту та документом, що посвідчує особу.

Якщо вкладник не відвідав відділення банку в день закінчення терміну депозиту або на наступний день активи будуть переведені на депозитний рахунок типу «до запитання».

Якщо в договорі вказана можливість автоматичної пролонгації та вкладник згодний з цим, він може не відвідувати відділення банку.

Типова послідовність дій для процесу відкриття депозиту оператором фронт офісу наведена в графічному матеріалі за допомогою структурної схеми діяльності.

Призначенням даної підсистеми є зменшення часу оформлення депозитів шляхом зменшення кількості документів, що мають заповнюватися власноруч. При відкритті нового депозиту оператор здійснює пошук клієнта в базі даних банку. В разі наявності клієнта в базі даних дані автоматично переносяться до форми введення, а в разі відсутності оператор вносить дані до

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

форми введення. Також оператор вводить дані про депозит та обирає тип депозиту. Дані для представлення можливих типів депозитів беруться з довідників депозитів, що знаходяться в БД банку. Після введення всіх даних оператором фронт офісу дані про депозит зберігаються до бази даних банку та відправляються оператору бек офісу для підтвердження. Також оператор фронт офісу має можливість надрукувати заяву на відкриття депозиту та за необхідності анкету клієнта. Після підтвердження даних депозиту оператором бек офісу, оператор фронт офісу може надрукувати договір з даними, що беруться з бази даних банку, що дає можливість клієнту банку не витрачати час на заповнення договору власноруч, а лише поставити підпис.

### **1.1.2. Опис функціональної моделі оформлення депозиту**

Для проектування діаграми використання перш за все потрібно визначити акторів підсистеми, а потім дії у системі, що може виконувати кожен з них. Акторами у системі є:

- оператор фронт офісу;
- оператор бек офісу.

Оператор фронт офісу, котрий має можливість перегляду відкритих за сьогодні договорів, всіх підтверджених договорів, відкриває нові депозити (вводить дані про клієнта, обирає тип депозита зі списку існуючих, вводить дані про депозит).

Документи, які може друкувати оператор фронт офісу:

- анкета клієнта;
- заява на відкриття депозиту;
- депозитний договір.

Оператор бек офісу, котрий підтверджує дані про оформлення депозиту, введені оператором фронт офісу. Структурна схема варіантів використання наведена в графічному матеріалі.

## 1.2. Огляд наявних аналогів

Нині існують АБС, що містять подібні підсистеми. В ході пошуку схожих за функціональністю депозитних модулів було виявлено чотири системи АБС, що містять схожі функції:

- Б2 [3];
- Грант [4];
- ИСАОД [5];
- Scrooge [6].

Проведемо аналіз існуючого функціоналу, що присутній в аналогах. Результати порівняння наведено в таблиці 1.1.

Таблиця 1.1 Результати порівняння наявних аналогів

Аналоги	Б2	Грант	ИСАОД	Scrooge
Функції				
Введення даних про клієнта	+	+	+	+
Вибір депозиту зі списку доступних депозитів	+	+	+	+
Введення даних депозиту	+	+	+	+
Друк депозитного договору	+	+	+	+
За необхідності друк анкети клієнта під час оформлення	+	–	–	+

## Продовження таблиці 1.1

Побудова діаграми на основі частотності вибору типу депозиту.	—	—	—	—
---	---	---	---	---

Як видно з порівняння представлених в таблиці 1.1 даних, всі продукти мають майже однаковий функціонал, але жоден з них не має можливості побудови діаграми на основі частотності вибору типу депозиту для аналізу існуючих депозитних продуктів.

### 1.3. Концепція депозитного модуля

#### 1.3.1. Опис результатів вивчення об'єкту автоматизації

##### 1.3.1.1. Уявлення про об'єкт автоматизації

Об'єктом автоматизації у дипломній роботі є процес оформлення депозита, який складається з заповнення договору, з подальшим підтвердженням та друком договору та заяви клієнта.

В даний час існує декілька подібних модулів в автоматизованих банківських системах Грант, ИСАОД та ін[3-6]. Основною проблемою використання існуючих рішень є відсутність перегляду статистики по існуючим депозитним продуктам, а також неповнота функціоналу та застаріли технології, що використовуються в деяких з них.

Основною вимогою для депозитного модуля є можливість інтеграції в автоматизовану банківську систему, забезпечення повноти функціоналу та надання можливості перегляду статистики по існуючим депозитним продуктам.

**1.3.1.2. Тенденції розвитку об'єкту автоматизації**

В даний час розвиток автоматизації процесів відбуваються у напрямку спрощення оформлення для клієнта і працівників банку та забезпечення оформлення депозитів шляхом заміни рукописних документів електронними, що формуються на основі введених даних при відкритті депозиту.

**1.3.2. Опис та оцінка переваг і недоліків розроблених альтернативних варіантів концепції створення модуля**

Основною вимогою до розроблюваної підсистеми є можливість інтеграції в автоматизовані банківські системи.

Альтернативним підходом до проектування є побудова окремої системи для оформлення та роботи з депозитами. Основною перевагою реалізації альтернативної концепції є порівняно менша складність реалізації, але дана концепції суперечить основній меті розробки – можливості інтеграції даної системи до вже існуючих автоматизованих банківських систем.

**1.3.3. Порівняльний аналіз вимог користувача до підсистеми і варіантів концепції підсистеми на предмет задоволення вимог користувача**

Для порівняння запропоновані дві альтернативні концепції, які є протилежними за суттю. Основна концепція, яка реалізує процес оформлення депозиту, як модуля, задовольняє вимогам користувача. Даний підхід є складніший в реалізації, оскільки потребує розроблювати програмний продукт з архітектурою, що дозволить легке інтегрування модуля до будь-якої автоматизованої банківської системи, але є більш ефективним для замовників тому, що вони можуть швидко почати використання модуля.

Оскільки основною вимогою є можливість функціонування, як частина існуючих банківських системах, тобто функціонування, як частина автоматизованої банківської системи тому, що для коректної роботи депозитного модуля необхідний функціонал інших модулів АБС(наприклад,

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

функціонал переводу коштів з одного рахунку на інший). Тоді альтернативна концепція буде їй суперечити вже на рівні проектування даної системи. Тому альтернативна концепція не може бути прийнятою.

#### **1.3.4. Очікувані результати та ефективність реалізації обраного варіанту концепції**

Реалізація депозитного модуля забезпечить збільшення швидкості та надійності роботи відділу банку з оформлення депозитів за рахунок автоматизації процесу оформлення депозитів та зберігання даних про клієнтів. При автоматизації даного процесу будуть досягнуті наступні цілі:

– для клієнта:

1. зменшення часу, який клієнт повинен витратити для того, щоб оформити депозит;
2. уникнення ситуацій введення некоретних даних клієнта або даних про рахунки для повернення коштів.

– для банку:

1. зменшення часу на обслуговування однієї людини оператором фронт офісу;
2. збільшення кількості клієнтів, що можуть бути обслужені за певний період часу;
3. підвищення зручності користування системою працівниками банку;
4. пониження ймовірності помилок при оформленні документів працівниками банку.

#### **1.3.5. Орієнтовний план реалізації обраного варіанту концепції підсистеми**

Будемо використовувати типовий цикл розробки програмного продукту при розробці депозитного модуля:

– розробка вимог;

					ДП ІС-5124.1181-с.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- проектування;
- програмна реалізація;
- тестування;
- впровадження системи;
- супровід.

#### ***1.3.6. Необхідні витрати ресурсів на розробку, введення в дію та забезпечення функціонування***

Дана система буде реалізовуватись однією людиною, що буде виконувати ролі бізнес-аналітика, дизайнера, розробника програмного забезпечення та тестувальника. Витрати на розробку даного модуля потрібно було б розраховувати з врахуванням часу, витраченого на розробку модуля з моменту розробки вимог до моменту інтеграції модуля до автоматизованої банківської системи.

#### ***1.3.7. Вимоги, що гарантують якість підсистеми***

Основним показником якості модуля є здатність виконувати функції, необхідні для працівників банку, що займаються оформленням депозитів. Також показником якості є стійкість, надійність, зручність інтерфейсу та сумісність з автоматизованими банківськими системами. Також для оцінки якості депозитного модуля необхідно провести опитування серед користувачів модуля – працівників банку для визначення якості системи за встановленим критерієм.

#### ***1.3.8. Умови приймання підсистеми***

Готовий депозитний модуль буде передано Замовнику у вигляді вихідних кодів програм (на електронному носії), а також вичерпну документацію до модуля, в якій буде зазначено опис необхідних для роботи компонентів та описано процес установки.

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

## 1.4. Постановка задачі

### 1.4.1. Призначення розробки

Призначенням депозитного модуля є підтримка малого та середнього бізнесу, аналіз популярності наявних депозитних продуктів шляхом перегляду побудованих діаграм, розрахунку кінцевої суми, отриманої від депозитного вкладу та знаходження дати повернення депозиту.

### 1.4.2. Цілі та задачі розробки

Основними цілями розробки є:

- зменшення часу на відкриття депозитів представникам малого бізнесу, середнього бізнесу та працівниками банку за рахунок автоматизації процесу формування документів на основі введених даних;
- пониження ймовірності помилок при оформленні документів працівниками банку за рахунок валідації форм введення даних та додаткової перевірки даних, збережених до бази даних працівниками бек офісу.

Для досягнення поставлених цілей мають бути вирішені такі задачі:

- створення функціоналу вибору для клієнта типу депозиту зі списку наявних депозитних продуктів;
- розробка функціоналу аналізу вибору наявних депозитних продуктів для працівників бек офісу банку, візуалізація аналітики;
- розробка бази даних для клієнтів, відкритих депозитів;
- створення функціоналу для пошуку клієнта в базі даних банку при оформленні депозита;
- розробка алгоритмів знаходження дати повернення депозиту, розрахунку прибутку отриманого від депозитного вкладу.



**Висновки до розділу**

В даному розділі було наведено опис предметного середовища, описано процес діяльності за допомогою діаграми IDEF0 та діаграми діяльності за стандартом UML, описано функціональну модель за допомогою діаграми варіантів використання за стандартом UML, розглянуто наявні аналоги та визначено переваги розроблюваного модуля, наведено концепцію депозитного модуля, сформульовано мету та цілі розроблюваного модуля, визначено задачі, які необхідно вирішити для досягнення цілей.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1. Вхідні дані підсистеми ВД

Первісні дані вводяться в систему оператором фронт офісу. Розглянемо структуру даних, яка потрібна для відкриття депозиту, а саме дані про клієнта, дані про тип депозитного продукту, дані про депозит.

Дані про клієнта:

- найменування клієнта банку – назва юридичної особи;
- в особі – прізвище ім'я по-батькові представника юридичної особи;
- код ідентифікації – код ПІН або код за ЄРДПОУ;
- місце реєстрації – місце реєстрації представника юридичної особи;
- номер телефону – номер телефону представника юридичної особи;
- номер банківського рахунку – номер банківського рахунку юридичної особи для списання коштів та повернення коштів після закінчення депозитного строку;
- номер банківського рахунку для депозиту – номер банківського рахунку для виплати відсотків по депозиту.

Дані про тип депозитного продукту:

- назва – назва типу депозитного продукту;
- автопродлонгація – наявність можливості автопродлонгації для даного депозитного продукту;
- можливість поповнення – наявність можливості поповнення для даного депозитного продукту;
- можливість дострокового розірвання – наявність можливості дострокового розірвання для даного депозитного продукту;
- відсоткова ставка – відсоткова ставка для даного депозитного продукту.

Дані про депозит:

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

- дата початку – дата початку депозитного строку;
- дата закінчення – дата закінчення депозитного строку;
- сума вкладу – сума вкладу при відкритті депозиту;
- валюта – валюта в якій відкривається депозит;
- клієнт – клієнт, що відкриває депозит;
- тип – тип обраного депозитного продукту.

## 2.2. Вихідні дані підсистеми ВД

Вихідними документами депозитного модуля є:

- анкета клієнта;
- заява на відкриття договору;
- депозитний договір.

У таблиці 2.1 наведений опис вихідних документів.

Таблиця 2.1 – Опис вихідних документів

№	Найменування	Кодове позначення	Реквізити
1	Анкета клієнта	АК	<ul style="list-style-type: none"> <li>- найменування клієнта банку;</li> <li>- в особі;</li> <li>- код ідентифікації;</li> <li>- місце реєстрації;</li> <li>- номер телефону;</li> <li>- номер банківського рахунку;</li> </ul>

## Продовження таблиці 2.1

			<ul style="list-style-type: none"> <li>- номер банківського рахунку для депозиту;</li> <li>- дата;</li> <li>- підпис клієнта</li> </ul>
2	Заява на відкриття договору	ЗВД	<ul style="list-style-type: none"> <li>- найменування клієнта банку;</li> <li>- в особі;</li> <li>- код ідентифікації;</li> <li>- місце реєстрації;</li> <li>- номер телефону;</li> <li>- номер;</li> <li>- банківського рахунку;</li> <li>- номер банківського рахунку для депозиту;</li> <li>- назва типу депозиту;</li> <li>- автопролонгація;</li> <li>- можливість поповнення;</li> </ul>

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 2.1

			<ul style="list-style-type: none"> <li>- можливість дострокового розірвання;</li> <li>- відсоткова ставка;</li> <li>- дата початку;</li> <li>- дата закінчення;</li> <li>- сума вкладу;</li> <li>- валюта;</li> <li>- дата;</li> <li>- підпис клієнта</li> </ul>
3	Депозитний договір	ДД	<ul style="list-style-type: none"> <li>- найменування клієнта банку;</li> <li>- в особі;</li> <li>- код ідентифікації;</li> <li>- місце реєстрації;</li> <li>- номер телефону;</li> <li>- номер банківського рахунку;</li> </ul>

## Продовження таблиці 2.1

			<ul style="list-style-type: none"> <li>- номер банківського рахунку для депозиту;</li> <li>- умови договору;</li> <li>- назва типу депозиту;</li> <li>- автопродлонгація;</li> <li>- можливість поповнення;</li> <li>- можливість дострокового розірвання;</li> <li>- відсоткова ставка;</li> <li>- дата початку;</li> <li>- дата закінчення;</li> <li>- сума вкладу;</li> <li>- валюта;</li> <li>- дата;</li> <li>- підпис працівника банку;</li> <li>підпис клієнта</li> </ul>
--	--	--	--

Змн.	Арк.	№ докум.	Підпис	Дата

### 2.3. Опис структури бази даних

Для зберігання даних створено три таблиці:

- DepositInfoes;
- Deposites;
- ClientInfoes.

Таблиця DepositInfoes містить інформацію про можливі типи депозитних продуктів, таблиця Deposites – інформацію про відкриті депозити, ClientInfoes – містить інформацію про клієнтів. Таблиці DepositInfoes та Deposites мають зв'язок один до одного або нуля, тобто Deposites обов'язково містить тип депозиту, а таблиця DepositInfoes може не містити посилання на Deposites. Аналогічний зв'язок мають таблиці Deposites та DepositInfoes.

Схема бази даних наведена в графічному матеріалі.

Таблиця 2.2 – опис структури бази даних

Назва Таблиці	Опис таблиці	Назва стовпчика	Опис стовпчика
DepositInfoes	Таблиця містить інформацію про всі наявні депозитні продукти	Id	Унікальний ідентифікатор депозитного продукту
		Name	Назва депозитного продукту
		Rate	Відсоткова ставка
		AutoRollover	Автопродлонгація
		PossibilityOfReplenishment	Можливість поповнення

## Продовження таблиці 2.2

		PossibilityOfTermination	Можливість дострокового розірвання
		Deposit_DepositId	Ідентифікатор в таблиці депозита
ClientInfoes	Таблиця містить інформацію про вкладників	Id	Унікальний ідентифікатор вкладника
		Name	Назва юридичної особи
		Representative	Представник юридичної особи
		IndentificationCode	Код ідентифікації вкладника
		RegistrationPlace	Місце реєстрації
		PhoneNumber	Телефонний номер
		BankAccount	Номер рахунку для повернення коштів
		ChangeDate	Дата зміни статусу депозита



## Продовження таблиці 2.2

		BankAccountForDP	Номер рахунку для виплати суми коштів нарахованих по відсоткам
		Deposit_DepositId	Ідентифікатор в таблиці депозита
Deposites	Таблиця містить інформацію про депозитні вклади	DepositId	Унікальний ідентифікатор депозита
		StartDepositDate	Дата початку депозитного строку
		EndDepositDate	Дата закінчення депозитного строку
		AmountOfDeposit	Сума депозитного вкладу
		Currency	Валюта вкладу
		Status	Статус депозита
		ClientInfoId	Ідентифікатор вкладника в таблиці ClientInfoes

## Продовження таблиці 2.2

		DepositInfoId	Ідентифікатор вкладника в таблиці DepositInfoes
		CreationDate	Дата створення
		AcceptionDate	Дата підтвердження
		PaymentDate	Дата виплати відсотків та повернення коштів
		AutoRolloverDate	Дата автопродлонгації
		TotalSum	Загальна сума коштів до повернення
		TotalRate	Сума коштів нарахованих по відсоткам
		Message	Повідомлення операторів

**Висновки до розділу**

В розділі з інформаційного забезпечення було зазначено вхідні дані, що потребує депозитний модуль для відкриття депозиту, структуру вихідних файлів, що формує модуль та зазначено структуру бази даних у вигляді діаграми та таблиці з детальним описом кожного з параметрів.

### 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1. Змістовна постановка задачі розрахунку результатів депозиту

Вклад коштів до депозитного рахунку відбувається за деякою відсотковою ставкою. Відсоткова ставка виступає, як певний стимул для вкладника для внесення коштів до депозитного рахунку. Вкладники, що вкладають кошти з метою зберігання їх обрають банк за стабільністю, а не за розміром відсоткової ставки. У випадку коли метою вкладника є прибуток, то він орієнтується на розмір відсоткової ставки. Величина відсоткової ставки визначається двома основними критеріями: сумою внеску та строком розміщення коштів.

Відсоткова ставка – відношення коштів отриманих у вигляді відсотка, до суми коштів кладених до депозитного рахунку. Дата виплати коштів нарахованих по відсоткам, дата повернення вкладених коштів, сума коштів нарахованих по відсотках описується у депозитному договорі. В основному відсоткова ставка стимулює вкладників до більш довгого зберігання коштів на депозитному рахунку, завдяки більшому прибутку.

Зміна відсоткової ставки може відбутися лише з погодженням з двома сторонами у разі зміни облікової ставки НБУ, при зміні кон'юнктури ринку, а також факт зміни відсоткової ставки по депозиту затверджується наказом по банку.

Виплата суми коштів нарахованої по відсоткам від депозитного вкладу здійснюється на рахунок вказаний в депозитному договорі. Даний рахунок може відповідати рахунку для виплати відсотків, а також може відрізнятися. Виплата суми коштів нарахованої по відсоткам можуть виплачуватися при закритті депозитного рахунку, періодично при використанні капіталізації відсотків, при внесенні коштів на депозит. Якщо вкладник бажає достроково повернути вклад, то сума коштів, що виплачується вкладнику є значно меншою завдяки зменшенню відсоткової ставки.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2. Математична постановка задачі розрахунку результатів депозиту

#### 3.2.1. Задача знаходження суми коштів по відсотковій ставці

Призначенням цієї задачі є розрахунок суми коштів нарахованих по відсотковій ставці.

Дано: дату початку депозитного строку, дату завершення депозитного строку, умови депозиту, відсоткову ставку, суму вкладу.

Потрібно знайти: суму коштів нарахованих по відсотковій ставці.

### 3.3. Обґрунтування методу розв'язання

#### 3.3.1. Задача знаходження суми коштів по відсотковій ставці

Нарахування відсотків здійснюється за двома типами відсоткової ставки: номінальної (розрахунок суми коштів нарахованих по відсоткам здійснюється на основі простих відсотків), фактичної (розрахунок суми коштів нарахованих по відсоткам здійснюється на основі складних відсотків).

### 3.4. Опис методів розв'язання

#### 3.4.1. Задача знаходження суми коштів по відсотковій ставці

Якщо використовується номінальна відсоткова ставка, сума коштів нарахованих по відсоткам розраховується, як добуток початкової суми вкладу без врахування коштів нарахованих по депозиту на множник нарахування. Сума коштів нарахованих по відсоткам депозиту розраховується за формулою:

$$S_{\pi} = \frac{S_{\text{в}} * r * N_{\text{д}}}{N_{\text{р}}} , \quad (1)$$

де  $S_{\pi}$  – сума відсотків за внеском,  $S_{\text{в}}$  – сума внеску,  $r$  – річна відсоткова ставка в вигляді десоткового дробу,  $N_{\text{д}}$  – кількість днів нарахування відсотків,  $N_{\text{р}}$  – кількість днів в році [7].

Якщо використовується складна відсоткова ставка, сума коштів нарахованих по відсоткам депозиту при використанні фактичної відсоткової ставки розраховується за формулою:

$$S_{\pi} = S_{\text{в}} * (1 + r)^n - S_{\text{в}} , \quad (2)$$

$$r = \frac{p * N_d}{N_p}, \quad (3)$$

де  $S_n$  – сума відсотків за внеском,  $S_v$  – сума внеску,  $r$  – відсоткова ставка в період капіталізації,  $p$  – річна відсоткова ставка в вигляді десяткового дробу,  $n$  – кількість періодів капіталізації,  $N_d$  – кількість днів нарахування відсотків,  $N_p$  – кількість днів в році [7] .

Наведемо алгоритм розрахування суми коштів нарахованої по відсоткам за формулою (1):

*КРОК 1.* отримати дату початку та дату кінця для конкретного депозиту;

*КРОК 2.* розрахувати кількість днів нарахування відсотків

*КРОК 3.* отримати відсоткову ставку по типу депозитного продукту, що відповідає даному депозиту.

*КРОК 4.* розрахувати кількість днів в році, в залежності від теперішнього року.

*КРОК 5.* розрахувати суму коштів нарахованої по відсоткам за внеском.

Наведемо алгоритм розрахування суми коштів нарахованої по відсоткам за формулою (2):

*КРОК 1.* отримати дату початку та дату кінця для конкретного депозиту;

*КРОК 2.* розрахувати кількість днів нарахування відсотків

*КРОК 3.* розрахувати кількість періодів капіталізації відсотків

*КРОК 4.* отримати відсоткову ставку по типу депозитного продукту, що відповідає даному депозиту.

*КРОК 5.* розрахувати кількість днів в році, в залежності від теперішнього року.

*КРОК 6.* розрахувати суму коштів нарахованої по відсоткам за внеском.

**Висновки до розділу**

В розділі з математичного забезпечення була сформульована змістовна та математична постановки задачі знаходження суми коштів нарахованих по відсотковій ставці та описані формули для розрахунку, та алгоритми знаходження шуканої величини, що реалізовані в модулі.

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

## 4. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1. Засоби розробки

При розробці підсистеми було обрано технологію WEB. Для створення клієнту було використано фреймворк AngularJS [8], що працює на мові JavaScript [9], а для сервера було використано мову C# [10], а саме C# Web Api 2 [11]. Для зберігання даних було використано базу даних Oracle 12c [12], а для взаємодії з нею фреймворк Entity Framework [13] для мови C# з використанням технології Code First [14].

C # підтримує дуже велику кількість інструментів і фреймворків. Основним IDE для розробки під платформу .Net є Microsoft Visual Studio, яка надає величезну кількість вбудованого функціоналу. Фреймворк .Net надає величезну кількість бібліотек для самих різних цілей.

Мова C# стрімко розвивається компанією Microsoft. Порівняно з деякими іншими мовами, новий функціонал, а також синтаксичні зміни відбуваються значно частіше. Також однією з значимих переваг є гнучкість мови та різноманітність типів додатків, що можуть бути створені мовою C#. Додатки можуть створюватися для операційної системи Windows, мобільні додатки за допомогою Xamarin, веб застосунки, ігри на основі ігрового движка Unity.

Аналогічні застосунки можна створювати за допомогою інших мов програмування, але це не можливо без сторонніх бібліотек та інструментів.

Entity Framework – спеціальна об'єктно-орієнтована технологія, що базується на .NET фреймворку для роботи з даними. Entity Framework надає більш високий рівень абстракції, ніж засоби ADO.NET, що дають можливість створювати підключення, виконувати SQL запити та інші операції з базами даних. Entity Framework дозволяє абстрагуватися від бази даних і працювати з даними незалежно від типу сховища. На рівні, запропонованого фреймворком, ми працюємо з об'єктами, а не з таблицями, ключами, індексами, що властиво традиційній роботі з базою даних[13].

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Основним поняттям Entity Framework - ку є сутність. Сутність – клас, що містить дані відповідної таблиці. При використанні даного фреймворку фактично використовується робота з класами замість роботи з запитом до бази даних.

Зв'язки між сутностями можуть бути: один-до-багатьох, один-до-одного, один-до-одного або нуля і багато-до-багатьох. Даний зв'язок встановлюється таким же чином, як і в базі даних.

Не менш важливою є Entity Data Model. Дана модель відповідає за зіставлення таблиць бази даних з сутностями, якими оперує сам фреймворк. Entity Data Model складається з трьох рівнів: концептуального, рівень сховища і рівень зіставлення (маппінга)[15].

Під час розробки програмного продукту було використано спосіб взаємодії Code First з базою даних, запропонований Entity Framework-ом.

Основною ідеєю Code First метода взаємодії з базою даних є створення класів сутностей, що будуть відповідати відповідним таблицям з бази даних та зберігатися в БД. На основі даних класів сутностей Entity Framework генерує базу даних (за необхідності), таблиці в базі даних та відповідні зв'язки між таблицями.

AngularJS – opensource JavaScript-фреймворк, що використовує шаблон проектування MVC. На представленні використовуються директиви фреймворку AngularJs, що надають самий різний функціонал. Також є можливість створення власних директив, та використанні їх, як звичайних html-тегів, властивостей html-тегів. Крім цього вся логіка роботи сторінки відокремлена в контроллер, та використовується при необхідності через відповідні директиви фреймворку. Дане розподілення функціоналу полегшує процес тестування і масштабування веб-застосунків. Також особливістю даного фреймворку є двійне зв'язування, що дає можливість змінювати дані моделі при зміні відповідних даних на веб застосунку та навпаки, при зміні



даних в моделі дані автоматично змінюються на веб застосунку( відбувається постійна синхронізація змінних в контролері, що відповідає представленню).

Також AngularJS пропонує такий функціонал, як Аїах запити , управління структурою дерева DOM( можна додавати та видаляти елементи з дерева DOM), анімація, шаблони( можна створювати власні шаблони та за допомогою директив вбудовувати їх як частину необхідних сторінок), маршрутизація( можна задавати маршрути при виклику яких, буде відбуватися виклик відповідного часткового представлення та використовуватися відповідний контроллер). Завдяки цьому AngularJs використовується у багатьох існуючих веб-застосунках.

JavaScript – це мова високого рівня без строгої типизації. Вона не дає можливості роботи з пам'ятю, як мова C++, оскільки спочатку вона розроблювалась для роботи з браузером, де дані можливості були абсолютно не потрібні. На даний момент мова JavaScript використовується в різних напрямках. В основному JavaScript використовують для роботи з html сторінками, та взаємодіяти з користувачем сайту. За допомогою мови JavaScript працюють зі структурою дерева DOM, додають нові теги, видаляють теги, додають класи, ідентифікатори, стилі, властивості тегам, приховують елементи сторінки, створюють обробники подій на сторінках( натискання клавіш на клавіатурі, заповнення полів, натискання на кнопки, переміщення курсора та інше), працюють з cookie, local storage, взаємодіють з сервером через Аїах-запити завдяки чому можна отримувати та відправляти дані на сервер без перезавантаження сторінки.

Oracle Database - це об'єктно-реляційна система керування базами даних [11].

З переваг Oracle 12с можна виділити високу продуктивність та підтримку баз даних дуже великих розмірів. Також Oracle 12с надає можливість змінювати рівень завантаженості системи для деяких процесів та підтримує розподілений доступ і користування базами даних. Oracle 12с

дозволяє імпортувати існуючі бази даних в вихідному вигляді, що дозволяє зменшити час на інтеграцію бази даних до вже існуючих баз даних. Однією з важливих переваг використання бази даних Oracle є невеликі вимоги сервера, де буде встановлена база даних.

## **4.2. Вимоги до технічного забезпечення**

### **4.2.1. Загальні вимоги**

Для коректного функціонування підсистеми було виділено наступні вимоги до технічного забезпечення.

До складу технічних засобів повинні входити:

- 1) комп'ютер з такою конфігурацією:
  - а) процесор з тактовою частотою не нижче 2 ГГц;
  - б) достатній об'єм оперативної пам'яті (не менше 4 Гб);
  - с) підключення до мережі Інтернет;
  - д) інші складові можуть мати будь-які параметри, тому що вони незначним чином впливають на роботу програми.
- 2) додатково має бути встановлене таке програмне забезпечення:
  - а) 64 розрядна операційна система Windows 10;
  - б) база даних Oracle Database 12c Express Edition;
  - с) .Net Framework 4.5 і вище;
  - д) Підтримка мови JavaScript в браузері.
- 3) комп'ютерна периферія, до складу якої входить:
  - а) монітор;
  - б) мишка;
  - с) клавіатура;
  - д) принтер.

### 4.3. Архітектура програмного забезпечення підсистеми “Депозитний модуль”

#### 4.3.1. Діаграма послідовності

На діаграмі послідовності зображено порядок взаємодії акторів з підсистемою, а також компонентів підсистеми між собою в ході процесу відкриття депозиту. Структурну схему послідовності наведено в графічному матеріалі.

#### 4.3.2. Діаграми класів

Оскільки було використано Entity Framework, то діаграма класів для сутностей показує об'єкти з якими ми будемо працювати для взаємодії з базою даних. Діаграма класів для сутностей є схожою на структурну схему бази даних, але деякі атрибути класів мають інші типи даних, проте вони конвертуються до вказаних в БД при записі. Діаграму класів, що відображають сутності для роботи з БД зображено на рисунку 4.1.

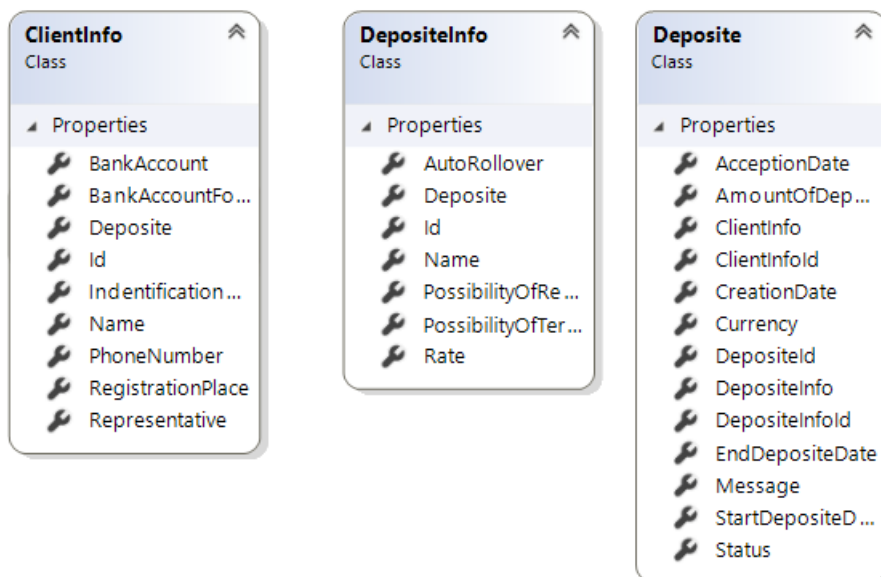


Рис.4.1. Діаграма класів для сутностей

Діаграма складається з 3 класів:

- «ClientInfo» – клас сутності, яка відповідає аналогічній таблиці ClientInfo в БД;
- «DepositInfo» – клас сутності, яка відповідає аналогічній таблиці DepositInfo в БД;
- «Deposit» – клас сутності, яка відповідає аналогічній таблиці Deposit в БД.

Для збереження даних до БД та повернення даних з сервера на клієнт було створенно класи-контролери для кожного з об'єктів сутностей. Якщо необхідно отримати дані з БД, необхідно зробити запит на відповідний метод контроллера, аналогічно і для запису та редагування даних, а також для іншого функціоналу закладеного в бізнес-логіку підсистеми, наприклад формування документів.

Діаграму класів, які використовуються для обміну даними між клієнтом і сервером зображено на рисунку 4.2

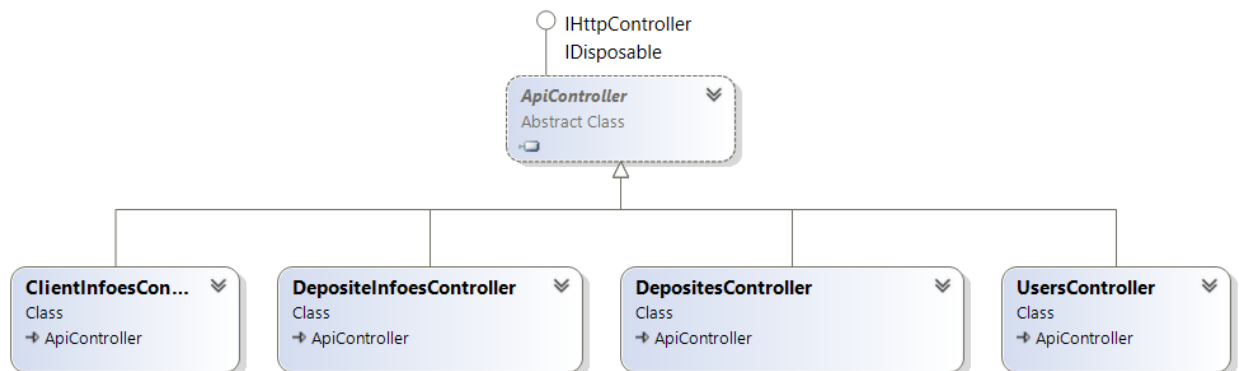


Рис.4.2. Діаграма класів для обміну даними між клієнтом та сервером

Діаграма складається з 4 класів:

- «DepositController» – клас для обміну даними між клієнтом і сервером об'єктами типу Deposit;
- «ClientInfosController» – клас для обміну даними між клієнтом і сервером об'єктами типу ClientInfo;

- «DepositInfoesController» – клас для обміну даними між клієнтом і сервером об'єктами типу DepositInfo;
- «UsersController» – клас для обміну даними між клієнтом і сервером об'єктами типу User.

У розгорнутому вигляді діаграма класів для обміну даними між клієнтом і сервером представлена в графічних матеріалах.

#### 4.3.2. Діаграма розгортання

При розробці підсистеми було використано клієнт-серверну архітектуру. Діаграма розгортання показує компоненти, що використовуються в підсистемі, та способи зв'язку компонентів між собою. На рисунку 4.3 зображено діаграму розгортання підсистеми “Депозитний модуль”.

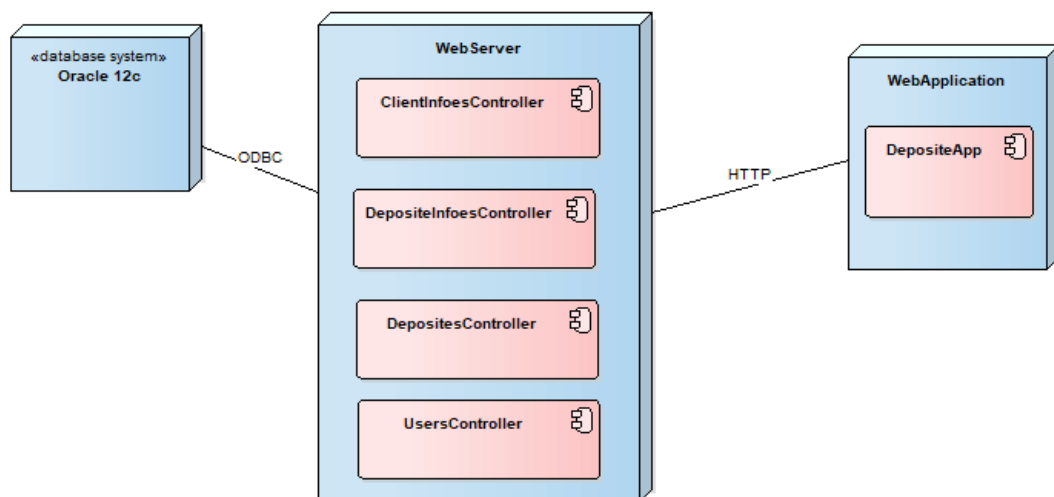


Рисунок 4.3. Діаграма розгортання підсистеми

#### 4.3.3. Специфікація функцій

Таблиця 4.1 – Специфікація функцій

Клас	Метод	Призначення	Повертає результат	Список параметрів	Семантика параметрів
FinalPayment	IsHoliday()	Перевіряє чи задана дата є в списку святкових днів	bool	DateTime date	Дата для перевірки

Продовження таблиці 4.1

	IsWeekEnd ( )	Перевіряє чи задана дата є вихідним днем	bool	DateTime date	Дата для перевірки
	VisYear ( )	Повертає кількість днів в заданому році	int	int year	Рік
	MonthDifference	Розраховує кількість днів між двома датами	int	DateTime IValue, DateTime rValue	Дата початку, дата кінця
	NumberOf DepositPeriods ( )	Розраховує кількість періодів капіталізації	int	DateTime startDate, DateTime endDate, CapitalizationType c	Дата початку депозитно го строку, дата завершенн я депозитно го строку, тип капіталізац ії
	SumOfRate ( )	Розраховує суму коштів нарахованих по відсоткам	double	string depositType, DateTime startDate,	Тип депозита, дата початку депозитно го строку,

Продовження таблиці 4.1

				DateTime endDate, int amountOfD eposite, int rate	дата завершенн я депозитно го строку, сума вкладу, відсоткова ставка
	GetTotalRa te ()	Присвоєння властивості значення суми коштів нарахованих по відсоткам для зовнішнього доступу	void	string depositTy pe, DateTime startDate, DateTime endDate, int amountOfD eposite, int rate	Тип депозита, дата початку депозитно го строку, дата завершенн я депозитно го строку,
	GetTotalSu m ()	Знаходження загальної суми для повернення коштів	void	string depositTy pe, DateTime startDate,	сума вкладу, відсоткова ставка, тип депозита, дата початку

Продовження таблиці 4.1

				DateTime endDate, int amountOfDeposit, int rate	депозитно го строку, дата завершен ня депозитно го строку, сума вкладу, відсоткова ставка
	GetNextWorkingDay ( )	Знаходження наступного робочого дня	DateTime	DateTime date	Дата для знаходжен ня
	GetPaymentDate ( )	Знаходження дати виплати	void	DateTime endDate	Дата закінчення депозитно го строку
DepositesController	GetStatistics()	Знаходження частотності вибору типу депозиту	Statistics	-	-
	GetGeneratedContract( )	Формування депозитного договору та повернення файлу на клієнт	HttpResponseMessage	int id	Ід депозиту



**Висновки до розділу**

В розділі програмного та технічного забезпечення було описано технології та засоби, що використовувались під час розробки депозитного модуля. Архітектура депозитного модуля була описана за допомогою діаграм класів, діаграми розгортання та наведено специфікацію функцій, що використовуються для розрахунку кінцевої суми коштів для повернення, суми коштів нарахованих по відсоткам, дату виплати відсотків та повернення коштів.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1. Керівництво користувача операторів фронт і бек офісу

Для того, щоб розпочати роботи з модулем — як з веб-застосунком необхідно авторизуватися в системі. Для авторизації необхідно ввести логін та пароль та натиснути на кнопку “Авторизуватися”. На рисунку 5.1 зображено вікно авторизації в веб застосунку.

#### Авторизація

Рисунок 5.1 – Авторизація в веб застосунку

Після авторизації ми переходимо на головну сторінку автоматизованої банківської системи та бачимо логін, під яким ми авторизувалися в правому верхньому кутку. На рисунку 5.2 головну сторінку автоматизованої банківської системи.

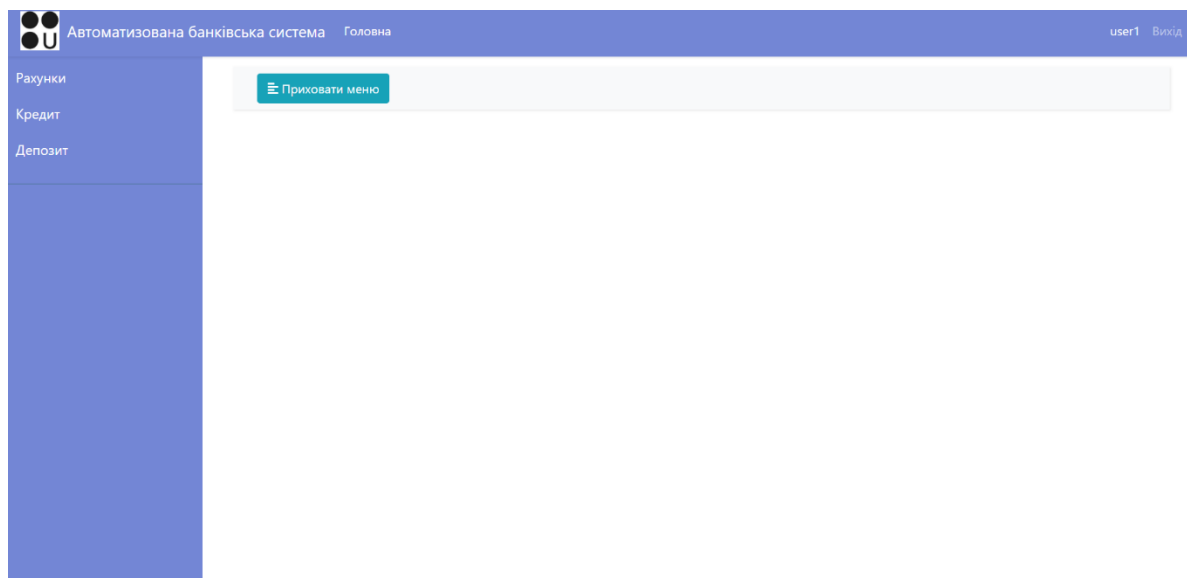


Рисунок 5.2 – Головна сторінка автоматизованої банківської системи

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Для того щоб перейти до роботи з депозитним модулем необхідно натиснути на кнопку “Депозит” в боковому меню з ліва сторінки. На рисунку 5.3 зображено сторінку депозитного модуля з авторизацією у ролі оператора фронт офісу.

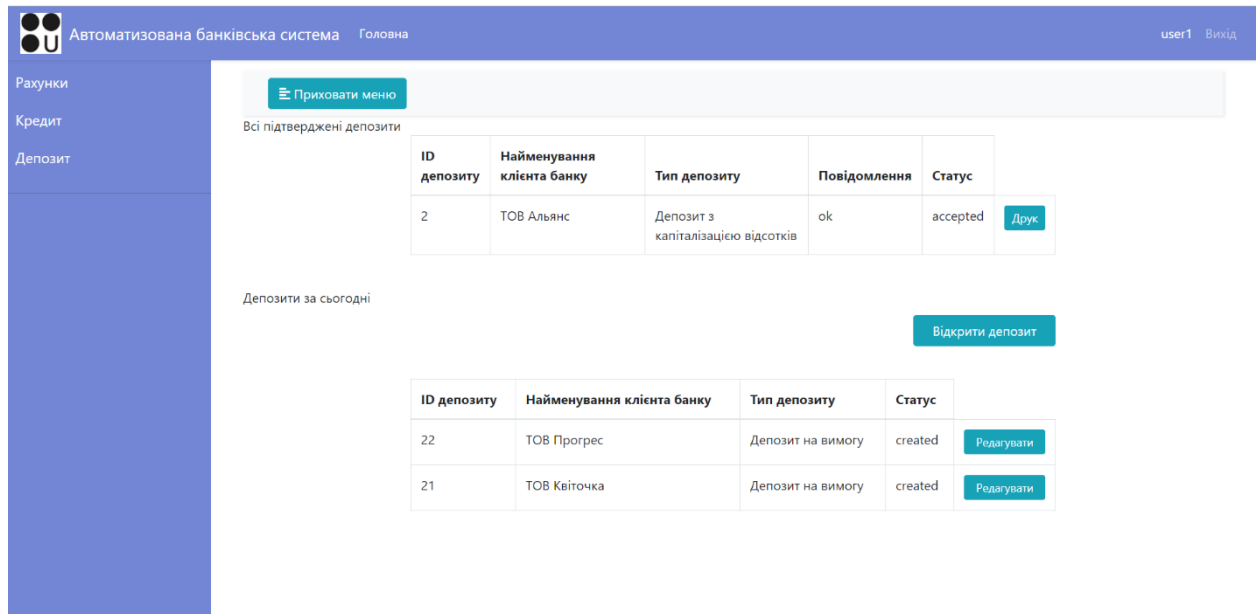


Рисунок 5.3 – Сторінка депозитного модуля оператора фронт офісу.

Для відкриття нового депозиту необхідно натиснути кнопку “Відкрити депозит”. На рисунку 5.4 зображено діалогове вікно для відкриття депозиту.

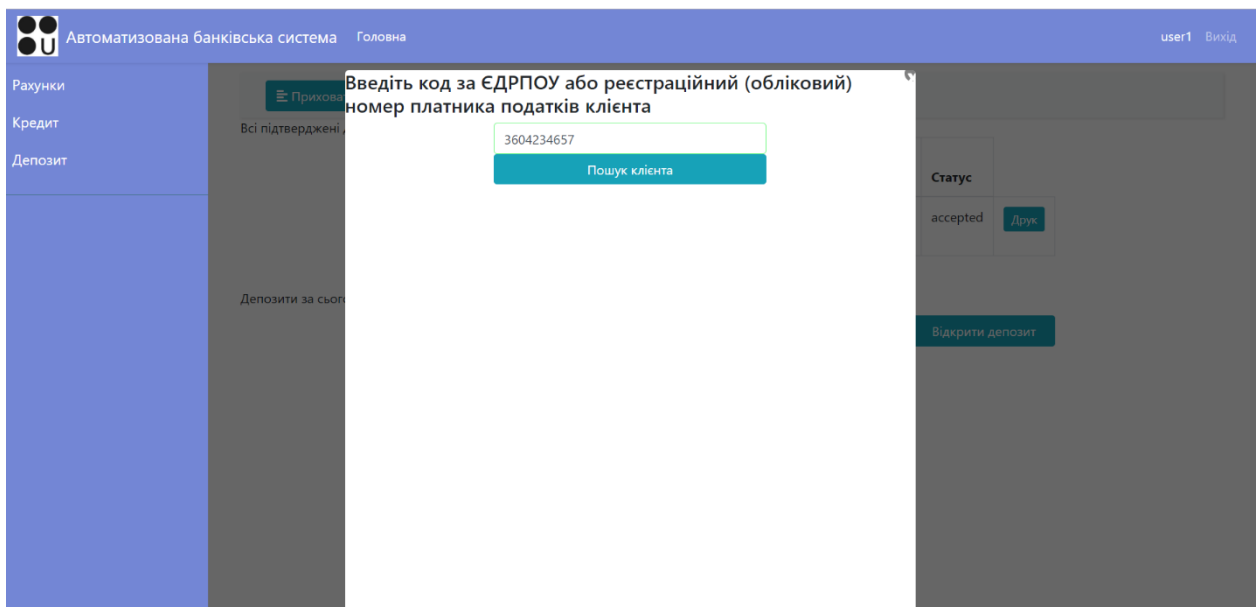


Рисунок 5.4 – Діалогове вікно відкриття депозиту.

Спочатку необхідно ввести код за ЕРДПОУ або реєстраційний номер платника податків та натиснути кнопку “Пошук клієнта” для пошуку

користувача за введеними даними в базі даних автоматизованої банківської системи. На рисунку 5.5 зображено діалогове вікно відкриття депозиту у разі коли клієнт є в базі даних автоматизованої банківської системи. Відповідні поля з даними про клієнта заповнюються автоматично.

Рисунок 5.5 –Діалогове вікно відкриття депозиту з заповненими даними про клієнта.

Після пошуку клієнта заповнюємо дані про депозит, а саме обираємо тип депозиту з списку доступних, вводимо початок депозитного строку та кінець, а також суму вкладу та обираємо валюту вкладу. Також ми бачимо відсоткову ставку та умови депозиту. Після чого натискаємо на кнопку “Зберегти депозит” для збереження даних про депозит до бази даних автоматизованої банківської системи. Введення даних про депозит зображено на рисунку 5.6.

Автоматизована банківська система Головна user1 Вихід

Рахунки  
Кредит  
Депозит

Тип депозиту: Депозит на вимогу

Особливості депозиту:

- Автопродовження: ☒
- Можливість поповнення: ☐
- Можливість дострокового розірвання: ☐
- Відсоткова ставка: 12

Початок депозитного строку: 02.06.2019

Кінець депозитного строку: 31.01.2020

Сума вкладу: 10000

Валюта: UAH

Зберегти депозит

Статус: accepted Друк

Відкрити депозит

Рисунок 5.6 –Введення даних про депозит.

Після збереження депозиту ми бачимо дві кнопки “Друк анкети клієнта” та “Друк заяви на відкриття депозиту”. При натисканні на них ми можемо сформувати відповідний pdf-документ, що завантажується і в подальшому може бути надрукований. Діалогове вікно з кнопками для друку зображено на рисунку 5.7.

Автоматизована банківська система Головна user1 Вихід

Рахунки  
Кредит  
Депозит

Натисність для друку договору

Друк анкети клієнта Друк заяви на відкриття депозиту

Депозит збережено

Статус: accepted Друк

Відкрити депозит

Рисунок 5.7 –Діалогове вікно з кнопками для друку.

Натиснемо на кнопку “Друк анкети клієнта” для формування клієнта, бачимо діалогове вікно з попередженням, про початок формування документа на кнопку “Ок” для закриття діалогового вікна попередження. На рисунку 5.8 зображено діалогове вікно попередження про початок друку документа.

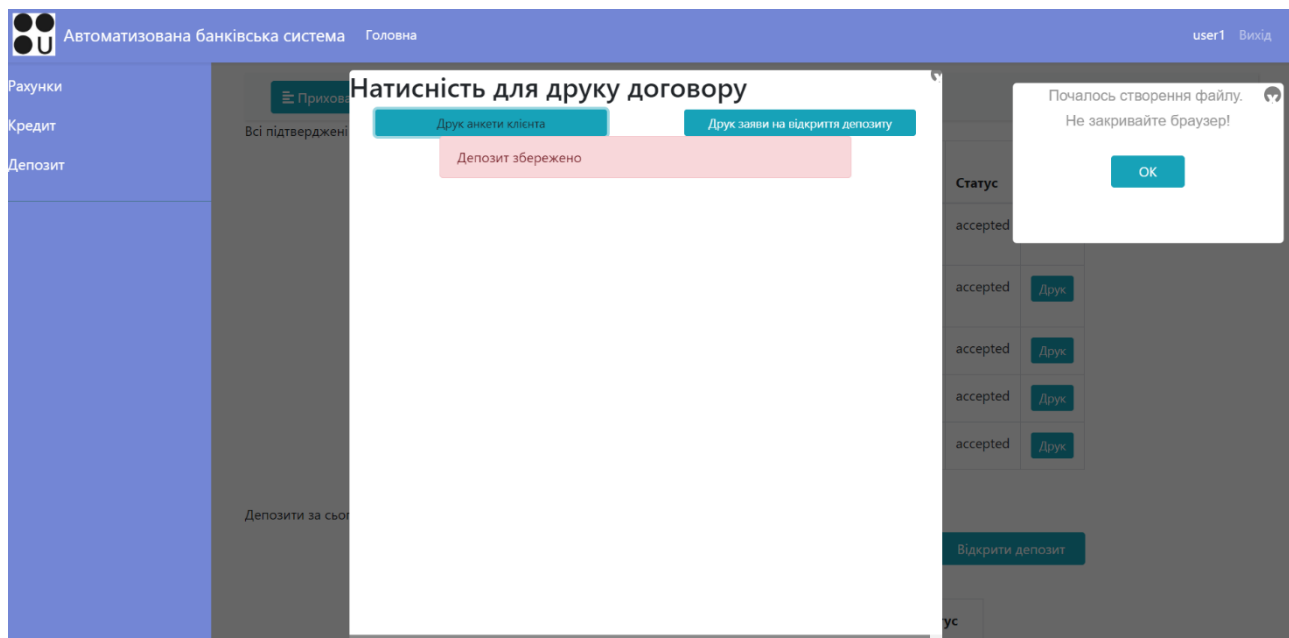


Рисунок 5.8 – Діалогове попередження про початок формування документа.

Для перегляду документа необхідно відкрити файл, що був щойно завантажений. Документ відкриється у новій вкладці браузера. Перегляд анкети клієнта зображено на рисунку 5.9.

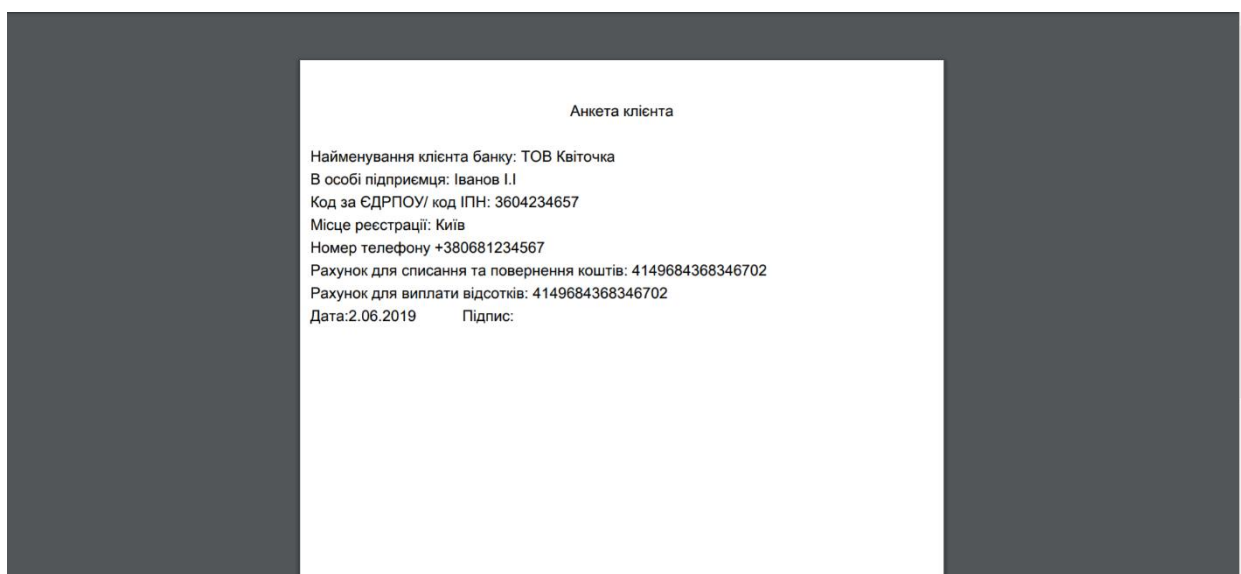


Рисунок 5.9 – Перегляд анкети клієнта.

					ДП ІС-5124.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

У разі коли клієнта не знайдено в базі даних автоматизованої банківської системи необхідно власноруч заповнити всі поля про клієнта та ввести дані про депозит. Введення даних про клієнта зображено на рисунку 5.10 та введення даних про депозит зображено на рисунку 5.11.

Автоматизована банківська система Головна user1 Вихід

Рахунки  
Кредит  
Депозит

Введіть код за ЄДРПОУ або реєстраційний (обліковий) номер платника податків клієнта

3607800134

Пошук клієнта

Клієнта не знайдено

Найменування клієнта банку: ТОВ Прогрес

В особі підприємця: Петров К.П.

код за ЄДРПОУ або реєстраційний (обліковий) номер платника податків клієнта: 3607800134

Місце реєстрації: Тернопіль

Номер телефону: +380681234067

Номер банківського: 3256789012340456

Статус: accepted Друк

Відкрити депозит

Регулювати

Рисунок 5.10 – Введення даних про клієнта.

Автоматизована банківська система Головна user1 Вихід

Рахунки  
Кредит  
Депозит

Тип депозиту: Депозит на вимогу

Особливості депозиту: Автопродлонгація: ☒ Можливість поповнення: ☐ Можливість дострокового розірвання: ☐ Відсоткова ставка: 12

Початок депозитного строку: 05.06.2019

Кінець депозитного строку: 31.10.2019

Сума вкладу: 1000

Валюта: UAH

Зберегти депозит

Статус: accepted Друк

Відкрити депозит

Регулювати

Рисунок 5.11 –Введення даних про депозит.

Після відкриття депозиту ми можемо переглянути список депозитів відкритих за сьогодні в нижній частині сторінки у вигляді таблиці. Таблиця відкритих за сьогодні депозитів зображена на рисунку 5.12.

					ДП ІС-5124.1181-с.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

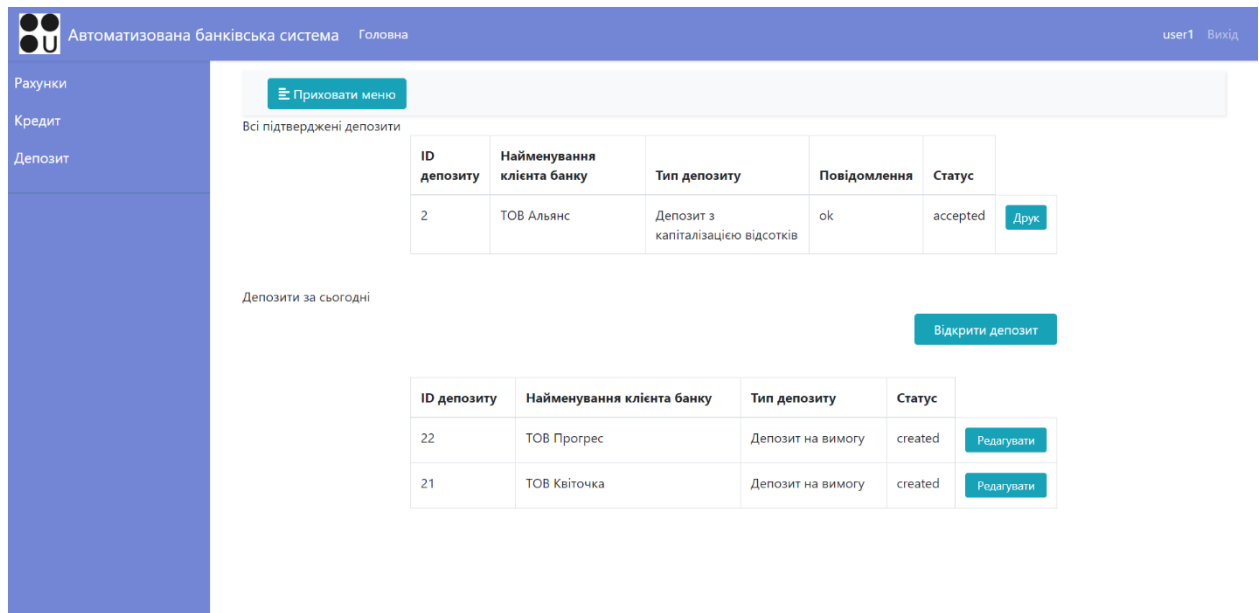


Рисунок 5.12 – Таблиця відкритих за сьогодні депозитів.

Після аторизації до системи, як оператор бек офісу ми бачимо сторінку, що забражена на рисунку 5.13.

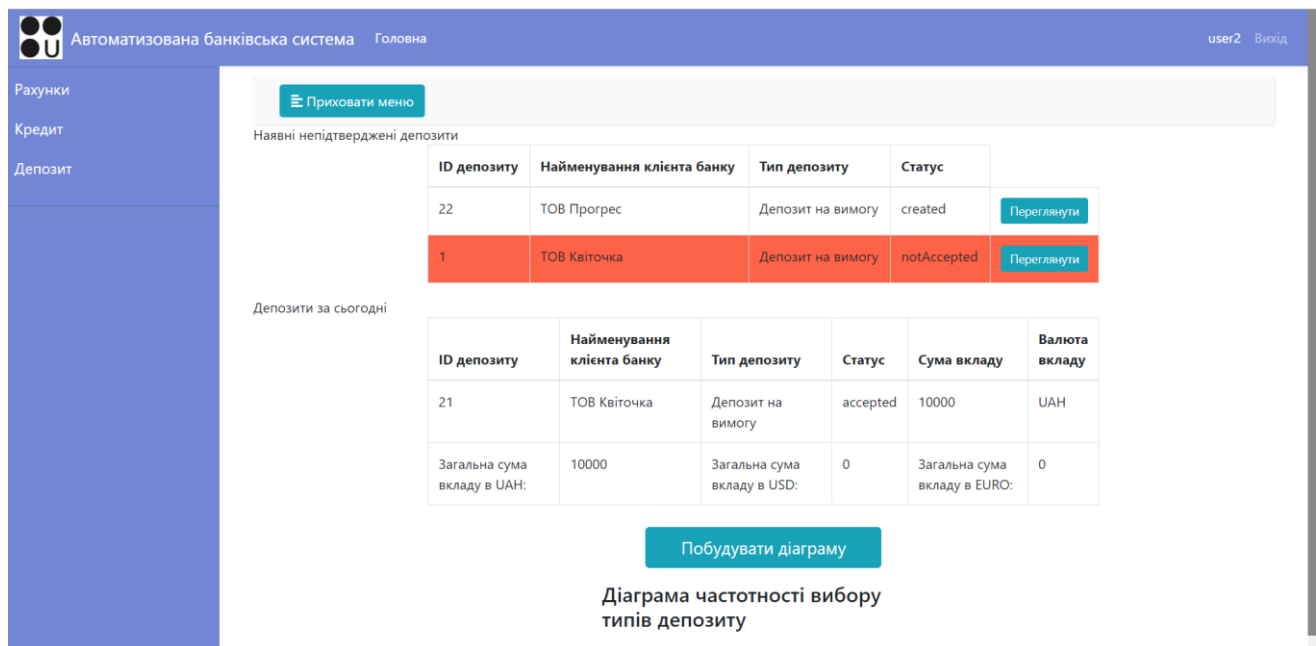


Рисунок 5.13 –Сторінка депозитного модуля після авторизації, як оператор бек офісу.

Для перегляду депозиту необхідно натиснути кнопку “Переглянути” навпроти необхідного депозиту. Для підтвердження депозиту необхідно ввести коментар та натиснути кнопку підтвердити у діалоговому вікні



перегляду даних про депозит. Діалогове вікно перегляду даних про депозит зображено на рисунку 5.14.

The dialog box displays the following information:

- Особливості депозиту:**
  - Автопродовження: ☒
  - Можливість поповнення: ☐
  - Можливість дострокового розірвання: ☐
  - Відсоткова ставка: 12
- Початок депозитного строку:** 15.05.2019
- Кінець депозитного строку:** 25.09.2019
- Сума вкладу:** 100000
- Валюта:** UAH
- Коментар:** Підтверджую

Buttons: Підтвердити, Не підтверджувати

Рисунок 5.14 – Діалогове вікно перегляду даних про депозит.

Після підтвердження депозиту всі підтверджені депозити за сьогодні можна переглянути в нижній частині сторінки в таблиці депозитів за сьогодні. Також в даній таблиці відображається інформація про загальну суму вкладів по кожній із валют. Таблиця підтверджених депозитів за сьогодні зображена на рисунку 5.15.

The table displays the following data:

ID депозиту	Найменування клієнта банку	Тип депозиту	Статус	Сума вкладу	Валюта вкладу
22	ТОВ Прогрес	Депозит на вимогу	created		
1	ТОВ Квіточка	Депозит на вимогу	accepted	100000	UAH
21	ТОВ Квіточка	Депозит на вимогу	accepted	10000	UAH
Загальна сума вкладу в UAH:	110000	Загальна сума вкладу в USD:	0	Загальна сума вкладу в EURO:	0

Buttons: Приховати меню, Переглянути, Побудувати діаграму

Діаграма частотності вибору типів депозиту

Рисунок 5.15 – Таблиця підтверджених депозитів за сьогоднішній день.

Якщо після перегляду депозиту оператор бек офісу бажає не підтверджувати депозит у діалоговому вікні перегляду даних про депозит він має ввести відповідний коментар та натиснути на кнопку “Не підтверджувати”. Діалогове вікно з відповідним коментарем зображено на рисунку 5.16.

Рисунок 5.16 – Діалогове вікно перегляду депозиту з відповідним коментарем.

Після того, як оператор бек офісу натиснув на кнопку “Не підтверджувати” і ввів відповідний коментар, даний депозит позначається червоним кольором у таблиці непідтверджених депозитів з відповідним статусом. Таблиця непідтверджених депозитів зображена на рисунку 5.17.

Автоматизована банківська система Головна user2 Вихід

Рахунки  
Кредит  
Депозит

Приховати меню

Відсутні непідтверджені депозити

ID депозиту	Найменування клієнта банку	Тип депозиту	Статус
22	ТОВ Прогрес	Депозит на вимогу	notAccepted

Депозити за сьогодні

ID депозиту	Найменування клієнта банку	Тип депозиту	Статус	Сума вкладу	Валюта вкладу
1	ТОВ Квіточка	Депозит на вимогу	accepted	100000	UAH
21	ТОВ Квіточка	Депозит на вимогу	accepted	10000	UAH
Загальна сума вкладу в UAH:	110000	Загальна сума вкладу в USD:	0	Загальна сума вкладу в EURO:	0

Побудувати діаграму

Діаграма частотності вибору типів депозиту

Рисунок 5.17 – Таблиця непідтверджених депозитів.

Для перегляду частотності вибору існуючих типів депозитів необхідно натиснути на кнопку “Побудувати діаграму”. Діаграма частотності вибору типів депозиту зображено на рисунку 5.18.

Автоматизована банківська система Головна user2 Вихід

Рахунки  
Кредит  
Депозит

1	ТОВ Квіточка	Депозит на вимогу	accepted	100000	UAH
21	ТОВ Квіточка	Депозит на вимогу	accepted	10000	UAH
Загальна сума вкладу в UAH:	110000	Загальна сума вкладу в USD:	0	Загальна сума вкладу в EURO:	0

Побудувати діаграму

Діаграма частотності вибору типів депозиту

Депозит з капіталізацією відсотків: 0.25

Рисунок 5.18 – Діаграма частотності вибору типів депозиту.

Для формування депозитного договору після підтвердження депозиту оператором бек офісу, необхідно зі сторінки депозиту оператора фронт офісу натиснути на кнопку “Друк” навпроти необхідного запису таблиці підтверджених депозитів та натиснути на кнопку “Ок” для закриття діалогового

вікна попередження. Діалогове вікно після натискання кнопки “Друк” зображено на рисунку 5.19.

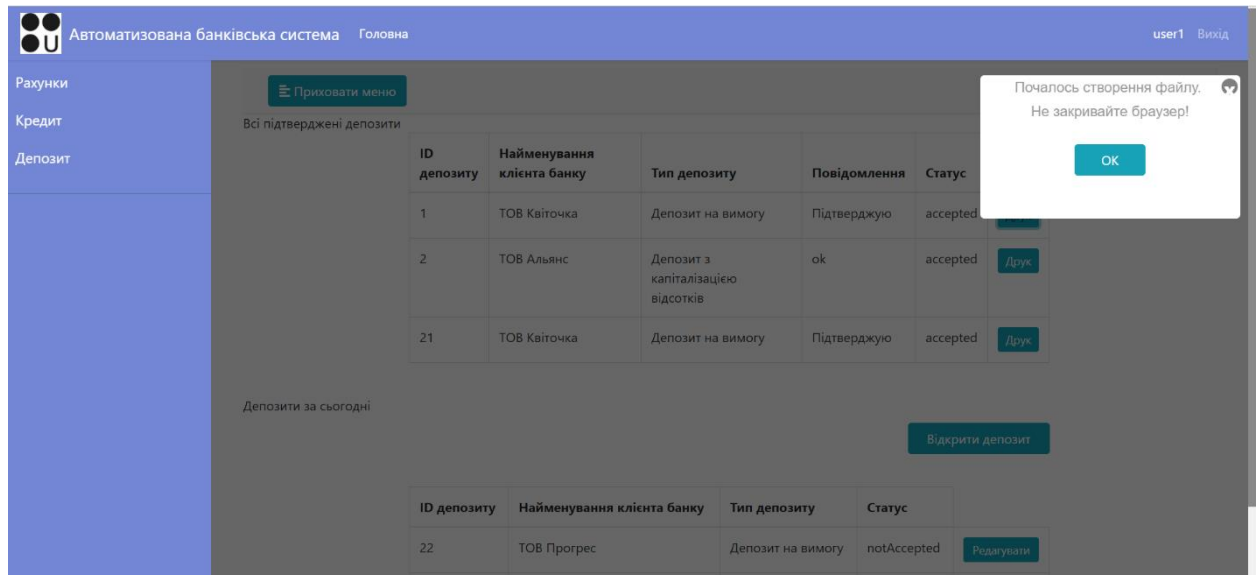


Рисунок 5.19 – Діалогове вікно попередження про створення файлу.

Для перегляду документа необхідно відкрити файл, що був щойно завантажений. В депозитного договорі вказано дані про клієнта, дані про депозит, дату повернення коштів та виплати суми депозиту, суму коштів нарахованих по відсоткам. Документ відкриється у новій вкладці браузера. Перегляд депозитного договору зображено на рисунку 5.20.

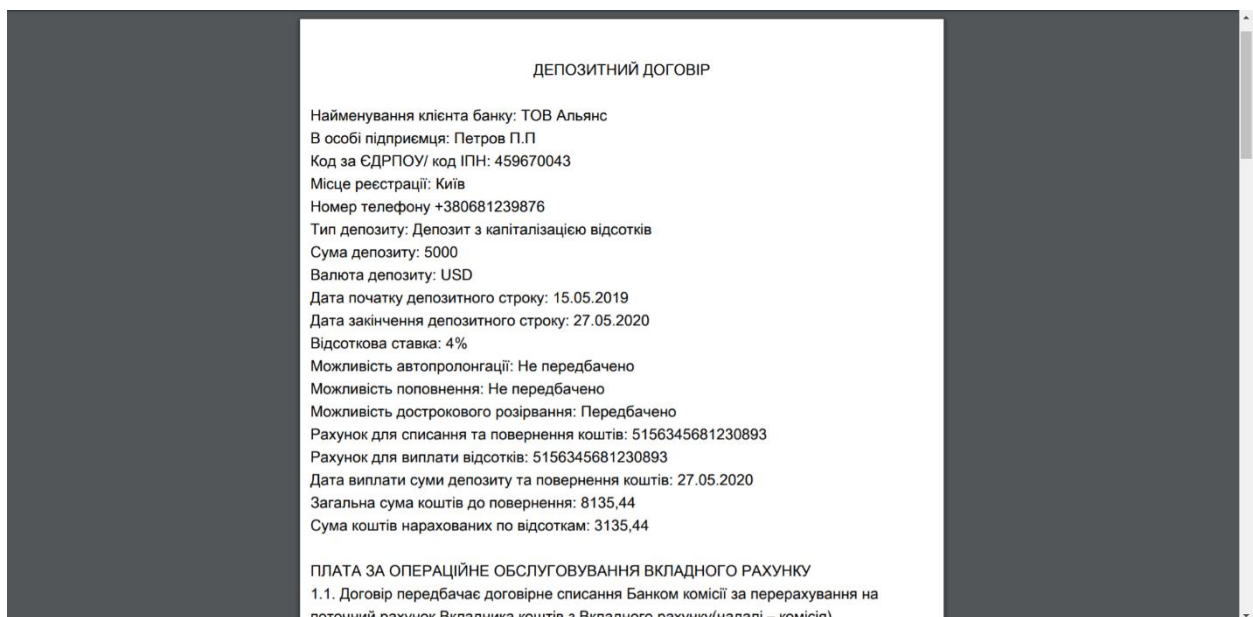


Рисунок 5.20 – Перегляд депозитного договору.

Для редагування даних про депозит необхідно натиснути кнопку “Редагувати” в таблиці відкритих за сьогодні депозитів. Таблиця відкритих за сьогодні депозитів з кнопками редагування зображена на рисунку 5.21.

Автоматизована банківська система Головна user1 Вихід

Рахунки  
Кредит  
Депозит

Всі підтверджені депозити

ID депозиту	Найменування клієнта банку	Тип депозиту	Повідомлення	Статус	
1	ТОВ Квіточка	Депозит на вимогу	Підтверджую	accepted	Друк
2	ТОВ Альянс	Депозит з капіталізацією відсотків	ok	accepted	Друк
21	ТОВ Квіточка	Депозит на вимогу	Підтверджую	accepted	Друк

Депозити за сьогодні

Відкрити депозит

ID депозиту	Найменування клієнта банку	Тип депозиту	Статус	
22	ТОВ Прогрес	Депозит на вимогу	notAccepted	Редагувати
21	ТОВ Квіточка	Депозит на вимогу	accepted	Редагувати

Рисунок 5.21 – Таблиця відкритих за сьогодні депозитів з кнопками редагування.

Після натискання кнопки редагування відкривається діалогове вікно з даними про депозит в якому ми можемо змінити деякі дані про депозит. На діалоговому вікні також присутні дві кнопки “Зберегти зміни” та “Завершити без змін”, що відповідно зберігають зміни та закривають діалогове вікно без змін. Діалогове вікно редагування зображено на рисунку 5.22.

Рисунок 5.22 – Діалогове вікно редагування.

Після натискання кнопки “Зберегти зміни” статус депозиту змінюється на створений. Стуси депозитів відкритих за сьогодні ми можемо переглянути в таблиці депозитів за сьогодні. Таблиця депозитів за сьогодні з оновленими статусами депозитів зображена на рисунку 5.23.

localhost:2570/DepositApp/Views/Index.html#/mainPage

user1 Вихід

Автоматизована банківська система

Головна

Рахунки

Кредит

Депозит

Всі підтверджені депозити

ID депозиту	Найменування клієнта банку	Тип депозиту	Повідомлення	Статус	
1	ТОВ Квіточка	Депозит на вимогу	Підтверджую	accepted	Друк
2	ТОВ Альянс	Депозит з капіталізацією відсотків	ok	accepted	Друк
21	ТОВ Квіточка	Депозит на вимогу	Підтверджую	accepted	Друк

Депозити за сьогодні

Відкрити депозит

ID депозиту	Найменування клієнта банку	Тип депозиту	Статус	
22	ТОВ Прорес	Депозит на вимогу	created	Редагувати
21	ТОВ Квіточка	Депозит на вимогу	accepted	Редагувати

Рисунок 5.23 – Таблиця депозитів за сьогодні з оновленими статусами депозитів.

## 5.2. Випробування програмного продукту підсистеми “Депозитний модуль”

### 5.2.1. Мета випробувань

Метою випробувань являється перевірка відповідності функцій модуля відкриття депозитів для малого та середнього бізнесу вимогам технічного завдання.

### 5.2.2. Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3. Результати випробувань

В процесі тестування була перевірена уся функціональність підсистеми та отримано результати, які еквівалентними до очікуваних. У наступних таблицях наведений перелік випробувань основного функціоналу (табл. 5.1 – 5.15).

Таблиця 5.1 – Перевірка відкриття форми відкриття депозиту

Мета тесту:	Перевірка функції «Відкриття форми відкриття депозиту»
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Відкрити новий депозит”

Продовження таблиці 5.1

Очікуваний результат:	Відкрилась форма для створення нового депозиту у діалоговому вікні
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу” з активною формою для створення нового депозиту у діалоговому вікні

Таблиця 5.2 – Перевірка пошуку клієнта в базі даних банку

Мета тесту:	Перевірка функції «Пошук клієнта в базі даних банку»
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	Код ідентифікації клієнта
Схема проведення тесту:	Натиснути кнопку “Пошук клієнта”
Очікуваний результат:	В формі у разі наявності клієнта у відповідних полях відображається інформація клієнта, а у разі відсутності поля пусті та код ідентифікації клієнта переноситься у наступне поле
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні

Таблиця 5.3 – Перевірка відкриття депозиту

Мета тесту:	Перевірка функції «Відкриття депозиту»
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні



## Продовження таблиці 5.3

Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Відкрити депозит”
Очікуваний результат:	Поля введення даних зникли, з’явилися 2 кнопки:”Друк анкети користувача”,”Друк заяви на відкриття депозиту”
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні

Таблиця 5.4 – Перевірка переходу до сторінки роботи з модулем користувача типу “Оператор фронт офісу” з форми відкриття депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Перехід до сторінки роботи з модулем користувача типу “Оператор фронт офісу” з форми відкриття депозиту»</b>
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Закрити”
Очікуваний результат:	Форма для створення нового депозиту зникла, відкрилась сторінка роботи з модулем користувача типу “Оператор фронт офісу”, додано запис в таблицю депозитів за сьогодні зі статусом “створений”

Продовження таблиці 5.4

Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор фронт офісу” з активною формою для створення нового депозиту у діалоговому вікні
---	--

Таблиця 5.5 – Перевірка валідності форми створення нового депозиту

Мета тесту:	Перевірка функції «Перевірка валідності форми створення нового депозиту»
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Відкрити депозит”
Очікуваний результат:	З’явилися відповідні помилки про хибність або відсутність введених даних
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні з вказаними помилками вводу

Таблиця 5.6 – Перевірка редагування даних депозиту

Мета тесту:	Перевірка функції «Редагування даних депозиту»
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”
Вхідні данні:	Id обраного депозиту
Схема проведення тесту:	Натиснути кнопку “Редагувати”

## Продовження таблиці 5.6

Очікуваний результат:	Відкрилась форма редагування обраного депозиту
Стан підсистеми після проведення випробувань:	Відкрита форма редагування депозиту у діалоговому вікні

## Таблиця 5.7 – Перевірка редагування депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Редагування депозиту»</b>
Початковий стан підсистеми	Відкрита форма редагування депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Підтвердити редагування”
Очікуваний результат:	Перехід до сторінки роботи з модулем користувача типу “Оператор фронт офісу” та збереження змін
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”

## Таблиця 5.8 – Перевірка валідності форми редагування депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Перевірка валідності форми редагування депозиту»</b>
Початковий стан підсистеми	Відкрита форма редагування депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит

Продовження таблиці 5.8

Схема проведення тесту:	Натиснути кнопку “Підтвердити редагування”
Очікуваний результат:	З’явилися відповідні помилки про хибність або відсутність введених даних
Стан підсистеми після проведення випробувань:	Відкрита форма редагування депозиту у діалоговому вікні

Таблиця 5.9 – Перевірка друку договору

Мета тесту:	Перевірка функції «Друк договору»
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”
Вхідні данні:	Id обраного депозиту
Схема проведення тесту:	Натиснути кнопку “Друк”
Очікуваний результат:	Відкрилось діалогове вікно з документом для друку
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу” з відкритим діалоговим вікном з документом для друку

Таблиця 5.10 – Перевірка відкриття форми підтвердження депозиту

Мета тесту:	Перевірка функції «Відкриття форми підтвердження депозиту»
Початковий стан підсистеми	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”

## Продовження таблиці 5.10

Вхідні данні:	Id обраного депозиту
Схема проведення тесту:	Натиснути кнопку “Переглянути”
Очікуваний результат:	Відкрита форма підтвердження депозиту у діалоговому вікні
Стан підсистеми після проведення випробувань:	Відкрита форма підтвердження депозиту у діалоговому вікні

Таблиця 5.11 – Перевірка підтвердження депозиту

Мета тесту:	Перевірка функції «Підтвердження депозиту»
Початковий стан підсистеми	Відкрита форма підтвердження депозиту у діалоговому вікні
Вхідні данні:	Коментар оператора бек-офісу
Схема проведення тесту:	Натиснути кнопку “Підтвердити”
Очікуваний результат:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”, дані про депозит збережено, додано запис в таблицю депозитів за сьогодні зі статусом “підтверджений”
Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”

Таблиця 5.12 – Перевірка відмови у підтвердженні

Мета тесту:	Перевірка функції «Відмова у підтвердженні»
Початковий стан підсистеми	Відкрита форма підтвердження депозиту у діалоговому вікні
Вхідні данні:	Коментар оператора бек-офісу
Схема проведення тесту:	Натиснути кнопку “Не підтверджувати”
Очікуваний результат:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”, дані про депозит збережено, додано запис в
	таблицю депозитів за сьогодні зі статусом “не підтверджений”
Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”

Таблиця 5.13 – Перевірка авторизації в системі

Мета тесту:	Перевірка функції «Авторизація в системі»
Початковий стан підсистеми	Відкрита сторінка авторизації
Вхідні данні:	Логін, пароль
Схема проведення тесту:	Натиснути кнопку “Авторизуватися”
Очікуваний результат:	Перехід на сторінку відповідно до ролі користувача

Продовження таблиці 5.13

Стан підсистеми після проведення випробувань:	Відкрита сторінка відповідно до ролі користувача
---	--

Таблиця 5.14 – Перевірка валідності форми авторизації

Мета тесту:	Перевірка функції «Перевірка валідності форми авторизації»
Початковий стан підсистеми	Відкрита сторінка авторизації
Вхідні данні:	Логін, пароль, що відстуні в БД, або пусті
Схема проведення тесту:	Натиснути кнопку “Авторизуватися”
Очікуваний результат:	Повідомлення про відповідну помилку входу
Стан підсистеми після проведення випробувань:	Відкрита сторінка авторизації

Таблиця 5.15 – Перевірка побудови діаграми частотності вибору типів депозиту

Мета тесту:	Перевірка функції «Побудова діаграми частотності вибору типів депозиту»
Початковий стан підсистеми	Відкрита сторінка депозитного модуля оператора бек офісу
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Побудувати діаграму”

Продовження таблиці 5.15

Очікуваний результат:	Побудована діаграма частотності вибору типів депозиту
Стан підсистеми після проведення випробувань:	Відкрита сторінка депозитного модуля оператора бек офісу

**Висновки до розділу**

У технологічному розділі було описано, як користуватися модулем у керівництві користувача, де було зображено покрокове використання основних функцій модуля, а також було проведене тестування функцій модуля на те чи відповідають вони функціональним вимогам.



## ЗАГАЛЬНІ ВИСНОВКИ

Під час виконання дипломної роботи було проаналізовано та модернізовано бізнес-процес оформлення депозиту. Бізнес-процес було описано за допомогою діаграми діяльності та діаграми бізнес-процесу в анотації IDEF0. Була розроблена функціональна модель, яка зображена за допомогою діаграми варіантів використання. Після модернізації бізнес-процес зчитує введені дані про депозит та вкладника і на основі них формує анкету клієнта, заяву на відкриття договору, депозитний договір.

Було розроблено структуру бази даних, діаграму структури БД та таблицю опису параметрів. Для розробки бази даних використовувалась технологія Oracle 12c.

Також було сформульовано змістовна та математична постановка задачі, наведено методи розв'язання та описано алгоритми за якими депозитний модуль буде розраховувати кінцеву суму нарахованих відсотків по відсотковій ставці.

Наведено опис технологій, що використовувалися при розробці депозитного модуля, та описана архітектура за допомогою діаграми послідовності, розгортання, класів. Також було наведено специфікацію функцій класів, що використовуються для реалізації алгоритмів описаних в розділі математичного забезпечення.

Також розроблено наведено керівництво користувача. Для підтвердження якості було проведено тестування функціоналу депозитного модуля.

В цілому проведена робота підтвердила досягнення цілей розробки депозитного модуля – зменшилась ймовірність помилок при оформленні депозитів та скоротився час оформлення депозитів для малого та середнього бізнесу.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Нові критерії підприємств (мікро, малі, середні, великі) – [Електроний ресурс] // Режим доступу: <https://zkg.ua/novi-kryteriji-pidpryjemstv-mikro-mali/>
2. Банківська енциклопедія / С.Г. Арбузов, Ю.В. Колобов, В.І. Міщенко, С.В. Науменкова. – К. : Центр наукових досліджень Національного банку України : Знання, 2011. – 504 с. – (Інституційні засади розвитку банківської системи України)
3. Б2 – АВТОМАТИЗИРОВАННАЯ БАНКОВСКАЯ СИСТЕМА – [Електроний ресурс] // Режим доступу: [https://csltd.com.ua/ru/products/core\\_banking](https://csltd.com.ua/ru/products/core_banking)
4. Автоматизированная банковская система – [Електроний ресурс] // Режим доступу: <http://www.banksoft.com.ua/project/bis-grant/>
5. Фронт-офисные системы – [Електроний ресурс] // Режим доступу: [http://www.ibis.ua/products/front\\_office/](http://www.ibis.ua/products/front_office/)
6. АБС SCROOGE– [Електроний ресурс] // Режим доступу: <http://lime-systems.com/products/abs-scrooge/>
7. Расчет процентов по вкладу: формула, как рассчитать?– [Електроний ресурс] // Режим доступу: <http://fingeniy.com/raschet-procentov-po-vkladu-formula-kak-rasschitat/>
8. Guide to AngularJS Documentation – [Електроний ресурс] // Режим доступу: <https://docs.angularjs.org/guide>
9. Офіційний сайт мови JavaScript – [Електроний ресурс] // Режим доступу: <https://www.javascript.com/>
10. C# programming guide– [Електроний ресурс] // Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
11. Get Started with ASP.NET Web API 2 (C#) – [Електроний ресурс] // Режим доступу: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>

12. Oracle Database 12c – [Електроний ресурс] // Режим доступу:  
<https://www.oracle.com/database/12c-database/>

13. Entity Framework Introduction Using C# - Part One – [Електроний ресурс] // Режим доступу: <https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/>

14. What is Code-First?– [Електроний ресурс] // Режим доступу:  
<http://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>

15. Введение в Entity Framework?– [Електроний ресурс] // Режим доступу: <https://metanit.com/sharp/entityframework/1.1.php>

## ДОДАТОК А

**Тексти програмного коду**

***Підсистема підтримки діяльності малого та середнього бізнесу у складі  
автоматизованої банківської системи “БАРС”***

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*20 арк, 304 Мб*

(Обсяг програми (документа), арк.,) Мб)

Київ – 2019 року

					КПІ ІС-5124.1181-с.ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Web.Http;
using System.Web.Http.Description;
using depositProject.Models;

namespace depositProject.Controllers
{
    public class DepositInfoesController : ApiController
    {
        private UsersDB db = new UsersDB();

        // GET: api/DepositInfoes
        public IQueryable<DepositInfo> GetDepositInfoes()
        {
            return db.DepositInfoes;
        }

        // GET: api/DepositInfoes/5
        [ResponseType(typeof(DepositInfo))]
        public IHttpActionResult GetDepositInfo(int id)
        {
            DepositInfo depositInfo = db.DepositInfoes.Find(id);
            if (depositInfo == null)
            {
                return NotFound();
            }

            return Ok(depositInfo);
        }

        // PUT: api/DepositInfoes/5
        [ResponseType(typeof(void))]
        public IHttpActionResult PutDepositInfo(int id, DepositInfo depositInfo)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            if (id != depositInfo.Id)
            {
                return BadRequest();
            }

            db.Entry(depositInfo).State = EntityState.Modified;

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!DepositInfoExists(id))
                {
                    return NotFound();
                }
            }
        }
    }
}

```

```

        else
        {
            throw;
        }
    }

    return StatusCode(HttpStatusCode.NoContent);
}

// POST: api/DepositInfoes
[ResponseType(typeof(DepositInfo))]
public IHttpActionResult PostDepositInfo(DepositInfo depositInfo)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    db.DepositInfoes.Add(depositInfo);
    db.SaveChanges();

    return CreatedAtRoute("DefaultApi", new { id = depositInfo.Id }, depositInfo);
}

// DELETE: api/DepositInfoes/5
[ResponseType(typeof(DepositInfo))]
public IHttpActionResult DeleteDepositInfo(int id)
{
    DepositInfo depositInfo = db.DepositInfoes.Find(id);
    if (depositInfo == null)
    {
        return NotFound();
    }

    db.DepositInfoes.Remove(depositInfo);
    db.SaveChanges();

    return Ok(depositInfo);
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}

private bool DepositInfoExists(int id)
{
    return db.DepositInfoes.Count(e => e.Id == id) > 0;
}
}

using System;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.IO;

```

```

using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Web.Http;
using System.Web.Http.Description;
using depositProject.Models;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace depositProject.Controllers
{
    [RoutePrefix("api/Deposites")]
    public class DepositesController : ApiController
    {
        private UsersDB db = new UsersDB();
        //return all unaccepted deposits
        // GET: api/Deposites
        public IQueryable<Deposit> GetDeposites()
        {
            var unacceptedDeposites = db.Deposites
                .Include(t => t.DepositInfo)
                .Include(p => p.ClientInfo);

            return unacceptedDeposites.Where(p => p.Status == "created");
        }
        [Route("AcceptedDeposites")]
        public IQueryable<Deposit> GetAcceptedDeposites()
        {
            var acceptedDeposites = db.Deposites
                .Include(t => t.DepositInfo)
                .Include(p => p.ClientInfo);

            return acceptedDeposites.Where(p => p.Status == "accepted");
        }
        [Route("NotAcceptedDeposites")]
        public IQueryable<Deposit> GetNotAcceptedDeposites()
        {
            var acceptedDeposites = db.Deposites
                .Include(t => t.DepositInfo)
                .Include(p => p.ClientInfo);

            return acceptedDeposites.Where(p => p.Status == "notAccepted");
        }
        [Route("AllDeposites")]
        public IQueryable<Deposit> GetAllDeposites()
        {
            //var allDeposites =

            return db.Deposites
                .Include(t => t.DepositInfo)
                .Include(p => p.ClientInfo);
        }
        [Route("ConfirmedTodayDeposites")]
        public IQueryable<Deposit> GetConfirmedTodayDeposites(DateTime date)
        {

            var confirmedTodayDeposites = db.Deposites
                .Include(t => t.DepositInfo)

```

```

.Include(p => p.ClientInfo);

return confirmedTodayDeposites.Where(p => p.AcceptionDate.Year == date.Year &&
p.AcceptionDate.Month==date.Month && p.AcceptionDate.Day==date.Day );
}
[Route("CreatedTodayDeposites")]
public IQueryable<Deposit> GetCreatedTodayDeposites(DateTime date)
{
    var createdTodayDeposites = db.Deposites
        .Include(t => t.DepositInfo)
        .Include(p => p.ClientInfo);
    var kind = date.Kind;
    return createdTodayDeposites.Where(p => p.CreationDate.Year == date.Year &&
p.CreationDate.Month == date.Month && p.CreationDate.Day == date.Day);
}

[Route("Statistics")]
public Statistics GetStatistics()
{
    Statistics stat = new Statistics();
    stat.CalculateFrequency();
    return stat;
}

// GET: api/Deposites/5
[ResponseType(typeof(Deposit))]
public IHttpActionResult GetDeposit(int id)
{
    Deposit deposit = db.Deposites.Find(id);
    if (deposit == null)
    {
        return NotFound();
    }

    return Ok(deposit);
}

// PUT: api/Deposites/5
[HttpPut]
[ResponseType(typeof(void))]
public IHttpActionResult PutDeposit(int id, Deposit deposit)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    if (id != deposit.DepositId)
    {
        return BadRequest();
    }

    db.Entry(deposit).State = EntityState.Modified;

    try
    {
        db.SaveChanges();
    }
    catch (DbUpdateConcurrencyException)

```



```

    {
        if (!DepositExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return StatusCode(HttpStatusCode.NoContent);
}

// POST: api/Deposites
[ResponseType(typeof(Deposit))]
public IHttpActionResult PostDeposit(Deposit deposit)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }
    db.Deposites.Add(deposit);
    db.SaveChanges();

    return CreatedAtRoute("DefaultApi", new { id = deposit.DepositId }, deposit);
}

// DELETE: api/Deposites/5
[ResponseType(typeof(Deposit))]
public IHttpActionResult DeleteDeposit(int id)
{
    Deposit deposit = db.Deposites.Find(id);
    if (deposit == null)
    {
        return NotFound();
    }

    db.Deposites.Remove(deposit);
    db.SaveChanges();

    return Ok(deposit);
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}

private string ChangeDateFormat(DateTime s)
{
    var temp="";
    if (s.Month<10)
    {
        temp = "0" + s.Month;
    } else

```

```

    {
        temp = "" + s.Month;
    }
    return s.Day + "." + temp + "." + s.Year;
}
private string ChangeConditionsView(bool info)
{
    if (info == true)
    {
        return "Передбачено";
    } else
    {
        return "Не передбачено";
    }
}

private bool DepositExists(int id)
{
    return db.Deposites.Count(e => e.DepositId == id) > 0;
}

```

```

[Route("GeneratedContract")]
public HttpResponseMessage GetGeneratedContract(int id)
{
    FinalPayment f = new FinalPayment();
    Deposit deposit = db.Deposites.Find(id);
    f.GetPaymentDate(deposit.EndDepositDate.ToLocalTime());
    f.GetTotalSum(deposit.DepositInfo.Name, deposit.StartDepositDate.ToLocalTime(),
deposit.EndDepositDate.ToLocalTime(), deposit.AmountOfDeposit, deposit.DepositInfo.Rate);
    deposit.PaymentDate = f.PaymentDate;
    if (deposit.DepositInfo.AutoRollover == true)
    {
        deposit.AutoRolloverDate = deposit.PaymentDate;
    }
    deposit.TotalRate = f.TotalRate;
    deposit.TotalSum = f.TotalSum;
    db.Entry(deposit).State = EntityState.Modified;

    try
    {
        db.SaveChanges();
    }
    catch (DbUpdateConcurrencyException)
    {
        throw new Exception();
    }
}

```

```

BaseFont baseFont = BaseFont.CreateFont(@"C:\ARIALUNI.TTF", BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
Font font = new Font(baseFont, 14, Font.NORMAL);
MemoryStream memoryStream = new MemoryStream();
Document document = new Document(PageSize.A4, 10, 10, 10, 10);
PdfWriter writer = PdfWriter.GetInstance(document, memoryStream);
document.Open();
Paragraph title = new Paragraph("\nДЕПОЗИТНИЙ ДОГОВІР \n\n", font)
{

```

					КПІ ІС-5124.1181-с.ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Alignment = Element.ALIGN_CENTER
    };
    document.Add(title);
    Paragraph para = new Paragraph("Найменування клієнта банку: " + deposit.ClientInfo.Name,
font);

    document.Add(para);
    Paragraph para6 = new Paragraph("В особі підприємця: " + deposit.ClientInfo.Representative,
font);

    document.Add(para6);
    Paragraph para1 = new Paragraph("Код за ЄДРПОУ/ код ІНН: " +
deposit.ClientInfo.IndentificationCode, font);
    document.Add(para1);
    Paragraph para2 = new Paragraph("Місце реєстрації: " + deposit.ClientInfo.RegistrationPlace,
font);

    document.Add(para2);
    Paragraph para3 = new Paragraph("Номер телефону " + deposit.ClientInfo.PhoneNumber, font);
    document.Add(para3);
    Paragraph para4 = new Paragraph("Тип депозиту: " + deposit.DepositInfo.Name, font);
    document.Add(para4);
    Paragraph para5 = new Paragraph("Сума депозиту: " + deposit.AmountOfDeposit, font);
    document.Add(para5);
    Paragraph para7 = new Paragraph("Валюта депозиту: " + deposit.Currency, font);
    document.Add(para7);
    Paragraph para8 = new Paragraph("Дата початку депозитного строку: " +
ChangeDateFormat(deposit.StartDepositDate.ToLocalTime()), font);
    document.Add(para8);
    Paragraph para9 = new Paragraph("Дата закінчення депозитного строку: " +
ChangeDateFormat(deposit.EndDepositDate.ToLocalTime()), font);
    document.Add(para9);
    Paragraph para10 = new Paragraph("Відсоткова ставка: " + deposit.DepositInfo.Rate + "%", font);
    document.Add(para10);
    Paragraph para11 = new Paragraph("Можливість автопродовження: " +
ChangeConditionsView(deposit.DepositInfo.AutoRollover), font);
    document.Add(para11);
    Paragraph para12 = new Paragraph("Можливість поповнення: " +
ChangeConditionsView(deposit.DepositInfo.PossibilityOfReplenishment), font);
    document.Add(para12);
    Paragraph para13 = new Paragraph("Можливість дострокового розірвання: " +
ChangeConditionsView(deposit.DepositInfo.PossibilityOfTermination), font);
    document.Add(para13);
    Paragraph para14 = new Paragraph("Рахунок для списання та повернення коштів: " +
deposit.ClientInfo.BankAccount, font);
    document.Add(para14);
    Paragraph para15 = new Paragraph("Рахунок для виплати відсотків: " +
deposit.ClientInfo.BankAccountForDP, font);
    document.Add(para15);
    Paragraph para18 = new Paragraph("Дата виплати суми депозиту та повернення коштів: " +
ChangeDateFormat(deposit.PaymentDate), font);
    document.Add(para18);
    Paragraph para19 = new Paragraph("Загальна сума коштів до повернення: " + deposit.TotalSum,
font);

    document.Add(para19);
    Paragraph para20 = new Paragraph("Сума коштів нарахованих по відсоткам: " +
deposit.TotalRate, font);
    document.Add(para20);
    Paragraph para17 = new Paragraph(Conditions, font);
    document.Add(para17);
    Paragraph para16 = new Paragraph("Дата:" + ChangeDateFormat(DateTime.Now) + " Підпис:",
font);

    document.Add(para16);

```

					КП ІС-5124.1181-с.ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

document.Close();
memoryStream.Close();
HttpResponseMessage result = new HttpResponseMessage(HttpStatusCode.OK)
{
    Content = new ByteArrayContent(memoryStream.ToArray())
};
//memoryStream.Close();
string pdfName = deposit.ClientInfo.Name + "_contract";
result.Content.Headers.ContentDisposition =
    new ContentDispositionHeaderValue("inline")
    {
        FileName = pdfName + ".pdf"
    };
result.Content.Headers.ContentType =
    new MediaTypeHeaderValue("application/octet-stream");
return result;
}
[Route("GeneratedStatement")]
public HttpResponseMessage GetGeneratedStatement(int id)
{
    Deposit deposit = db.Deposits.Find(id);
    BaseFont baseFont = BaseFont.CreateFont(@"C:\ARIALUNI.TTF", BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
    iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14,
iTextSharp.text.Font.NORMAL);
    MemoryStream memoryStream = new MemoryStream();
    Document document = new Document(PageSize.A4, 10, 10, 10, 10);
    PdfWriter writer = PdfWriter.GetInstance(document, memoryStream);
    document.Open();
    Paragraph title = new Paragraph("\nЗАЯВА НА ВІДКРИТТЯ ДЕПОЗИТУ \n\n", font)
    {
        Alignment = Element.ALIGN_CENTER
    };
    document.Add(title);
    Paragraph para = new Paragraph("Найменування клієнта банку: " + deposit.ClientInfo.Name,
font);
    document.Add(para);
    Paragraph para6 = new Paragraph("В особі підприємця: " + deposit.ClientInfo.Representative,
font);
    document.Add(para6);
    Paragraph para1 = new Paragraph("Код за ЄДРПОУ/ код ПІН: " +
deposit.ClientInfo.IndentificationCode, font);
    document.Add(para1);
    Paragraph para2 = new Paragraph("Місце реєстрації: " + deposit.ClientInfo.RegistrationPlace,
font);
    document.Add(para2);
    Paragraph para3 = new Paragraph("Номер телефону " + deposit.ClientInfo.PhoneNumber, font);
    document.Add(para3);
    Paragraph para4 = new Paragraph("Тип депозиту: " + deposit.DepositInfo.Name, font);
    document.Add(para4);
    Paragraph para5 = new Paragraph("Сума депозиту: " + deposit.AmountOfDeposit, font);
    document.Add(para5);
    Paragraph para7 = new Paragraph("Валюта депозиту: " + deposit.Currency, font);
    document.Add(para7);
    Paragraph para8 = new Paragraph("Дата початку депозитного строку: " +
ChangeDateFormat(deposit.StartDepositDate.ToLocalTime()), font);
    document.Add(para8);
    Paragraph para9 = new Paragraph("Дата закінчення депозитного строку: " +
ChangeDateFormat(deposit.EndDepositDate.ToLocalTime()), font);

```

					КПІ ІС-5124.1181-с.ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

document.Add(para9);
Paragraph para10 = new Paragraph("Відсоткова ставка: " + deposit.DepositInfo.Rate+"%", font);
document.Add(para10);
Paragraph para11 = new Paragraph("Можливість автопронгації: " +
ChangeConditionsView(deposit.DepositInfo.AutoRollover), font);
document.Add(para11);
Paragraph para12 = new Paragraph("Можливість поповнення: " +
ChangeConditionsView(deposit.DepositInfo.PossibilityOfReplenishment), font);
document.Add(para12);
Paragraph para13 = new Paragraph("Можливість дострокового розірвання: " +
ChangeConditionsView(deposit.DepositInfo.PossibilityOfTermination), font);
document.Add(para13);
Paragraph para14 = new Paragraph("Рахунок для списання та повернення коштів: " +
deposit.ClientInfo.BankAccount, font);
document.Add(para14);
Paragraph para15 = new Paragraph("Рахунок для виплати відсотків: " +
deposit.ClientInfo.BankAccountForDP, font);
document.Add(para15);

Paragraph para16 = new Paragraph("Дата:" + ChangeDateFormat(DateTime.Now)+" Підпис:",
font);

document.Add(para16);
document.Close();
memoryStream.Close();
HttpResponseMessage result = new HttpResponseMessage(HttpStatusCode.OK)
{
    Content = new ByteArrayContent(memoryStream.ToArray())
};

string pdfName = deposit.ClientInfo.Name+ "_statement";
result.Content.Headers.ContentDisposition =
    new ContentDispositionHeaderValue("attachment")
    {
        FileName = pdfName + ".pdf"
    };
result.Content.Headers.ContentType =
    new MediaTypeHeaderValue("application/octet-stream");
return result;
}
[Route("GeneratedClientForm")]
public HttpResponseMessage GetGeneratedClientForm(int id)
{
    Deposit deposit = db.Deposits.Find(id);
    BaseFont baseFont = BaseFont.CreateFont(@"C:\ARIALUNI.TTF", BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
    iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, 14,
iTextSharp.text.Font.NORMAL);
    MemoryStream memoryStream = new MemoryStream();
    Document document = new Document(PageSize.A4, 10, 10, 10, 10);
    PdfWriter writer = PdfWriter.GetInstance(document, memoryStream);
    document.Open();
    Paragraph title = new Paragraph("\nАнкета клієнта \n\n", font);
    title.Alignment = Element.ALIGN_CENTER;
    document.Add(title);
    Paragraph para = new Paragraph("Найменування клієнта банку: " + deposit.ClientInfo.Name,
font);
    document.Add(para);
    Paragraph para6 = new Paragraph("В особі підприємця: " + deposit.ClientInfo.Representative,
font);

```

					КПІ ІС-5124.1181-с.ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        document.Add(para6);
        Paragraph para1 = new Paragraph("Код за ЄДРПОУ/ код ПІН: " +
deposit.ClientInfo.IndentificationCode, font);
        document.Add(para1);
        Paragraph para2 = new Paragraph("Місце реєстрації: " + deposit.ClientInfo.RegistrationPlace,
font);

        document.Add(para2);
        Paragraph para3 = new Paragraph("Номер телефону " + deposit.ClientInfo.PhoneNumber, font);
        document.Add(para3);
        Paragraph para14 = new Paragraph("Рахунок для списання та повернення коштів: " +
deposit.ClientInfo.BankAccount, font);
        document.Add(para14);
        Paragraph para15 = new Paragraph("Рахунок для виплати відсотків: " +
deposit.ClientInfo.BankAccountForDP, font);
        document.Add(para15);
        Paragraph para16 = new Paragraph("Дата:" + ChangeDateFormat(DateTime.Now) + " Підпис:",
font);

        document.Add(para16);
        document.Close();
        memoryStream.Close();
        HttpResponseMessage result = new HttpResponseMessage(HttpStatusCode.OK)
        {
            Content = new ByteArrayContent(memoryStream.ToArray())
        };
        //memoryStream.Close();
        string pdfName = deposit.ClientInfo.Name + "_clientForm";
        result.Content.Headers.ContentDisposition =
            new ContentDispositionHeaderValue("attachment")
            {
                FileName = pdfName + ".pdf"
            };

        result.Content.Headers.ContentType =
            new MediaTypeHeaderValue("application/octet-stream");
        return result;
    }
}
namespace depositProject.Models
{
    public enum CapitalizationType
    {
        EveryMonth = 1,
        EveryQuarter = 3,
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Web;

namespace depositProject.Models
{
    public class ClientInfo
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]

```

```

        public int Id { get; set; }
        public string Name { get; set; }
        public string Representative { get; set; }
        public string IdentificationCode { get; set; }
        public string RegistrationPlace { get; set; }
        public string PhoneNumber { get; set; }
        public string BankAccount { get; set; }
        public DateTime ChangeDate { get; set; }
        public string BankAccountForDP { get; set; }
        public virtual Deposit Deposit { get; set; }
    }
}
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace depositProject.Models
{
    public class Deposit
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int DepositId { get; set; }
        public DateTime StartDepositDate { get; set; }
        public DateTime EndDepositDate { get; set; }
        public int AmountOfDeposit { get; set; }
        public string Currency { get; set; }
        public string Status { get; set; }
        public int ClientInfoId { get; set; }
        public int DepositInfoId { get; set; }
        public DateTime CreationDate { get; set; }
        public DateTime AcceptanceDate { get; set; }
        public DateTime PaymentDate { get; set; }
        public DateTime AutoRolloverDate { get; set; }
        public double TotalSum { get; set; }
        public double TotalRate { get; set; }
        public string Message { get; set; }
        public virtual ClientInfo ClientInfo { get; set; }
        public virtual DepositInfo DepositInfo { get; set; }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Web;

namespace depositProject.Models
{
    public class DepositInfo
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
        public string Name { get; set; }
        public int Rate { get; set; }
        public bool AutoRollover { get; set; }
        public bool PossibilityOfReplenishment { get; set; }
    }
}

```

```

        public bool PossibilityOfTermination { get; set; }
        public Deposit Deposit { get; set; }
    }
} using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web;

namespace depositProject.Models
{
    public class FinalPayment
    {
        public double TotalSum { get; set; }
        public DateTime PaymentDate { get; set; }
        public double TotalRate { get; set; }

        private bool IsHoliday(DateTime date)
        {
            string[] hds = { "01.01", "07.01", "08.03", "28.04", "29.04", "30.04", "01.05", "09.05", "16.06",
"17.06", "28.06", "16.07", "28.07", "24.08", "26.08", "14.10", "25.12", "30.12", "31.12" };

            foreach (var hd in hds)
            {
                if (date.Date.ToString("dd.MM", CultureInfo.InvariantCulture) == hd) return true;
            }
            return false;
        }
        private static bool IsWeekEnd(DateTime date)
        {
            if (date.DayOfWeek == DayOfWeek.Saturday || date.DayOfWeek == DayOfWeek.Sunday)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        private int VisYear(int year)
        {
            if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
            {
                return 366;
            }
            else
            {
                return 365;
            }
        }
        private int MonthDifference(DateTime lValue, DateTime rValue)
        {
            return (lValue.Month - rValue.Month) + 12 * (lValue.Year - rValue.Year);
        }
        private int NumberOfDepositPeriods(DateTime startDate, DateTime endDate, CapitalizationType c)
        {
            if (c == CapitalizationType.EveryMonth)
            {
                return MonthDifference(endDate, startDate);
            }
        }
    }
}

```



```

    }
    else
    {
        return MonthDifference(endDate, startDate) / 4;
    }
}

private double SumOfRate(string depositType, DateTime startDate, DateTime endDate, int
amountOfDeposit, int rate)
{
    int term = (endDate.Date - startDate.Date).Days;
    if (depositType == "Депозит на вимогу")
    {
        return Math.Round(((double)amountOfDeposit * ((double)rate / (double)100) * (double)term) /
VisYear(DateTime.Now.Year), 2);
    }
    else
    //if(depositType == "Депозит стандартний")
    {
        double monthRate = Math.Round(((double)rate / (double)100) * (double)term /
VisYear(DateTime.Now.Year), 4);
        if (NumberOfDepositPeriods(startDate, endDate, CapitalizationType.EveryMonth) == 0)
        {
            return Math.Round((double)amountOfDeposit * (1 + monthRate) - amountOfDeposit, 2);
        }
        else
        {
            return Math.Round((double)amountOfDeposit * Math.Pow((1 + monthRate),
NumberOfDepositPeriods(startDate, endDate, CapitalizationType.EveryMonth)) - amountOfDeposit, 2);
        }
    }
}

public void GetTotalRate(string depositType, DateTime startDate, DateTime endDate, int
amountOfDeposit, int rate)
{
    this.TotalRate = SumOfRate(depositType, startDate, endDate, amountOfDeposit, rate);
}

public void GetTotalSum(string depositType, DateTime startDate, DateTime endDate, int
amountOfDeposit, int rate)
{
    GetTotalRate(depositType, startDate, endDate, amountOfDeposit, rate);
    this.TotalSum = this.TotalRate + amountOfDeposit;
}

private DateTime GetNextWorkingDay(DateTime date)
{
    do
    {
        date = date.AddDays(1);
    } while (IsHoliday(date) || IsWeekEnd(date));
    return date;
}

public void GetPaymentDate(DateTime endDate)
{
    if (IsHoliday(endDate) != true && !IsWeekEnd(endDate))
    {
        this.PaymentDate = endDate;
    }
}

```

```

        else
        {
            this.PaymentDate = GetNextWorkingDay(endDate);
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;

namespace depositProject.Models
{
    public class Statistics
    {
        private UsersDB db = new UsersDB();

        public List<string> Labels { get; set; }
        public List<double> Data { get; set; }

        public void CalculateFrequency()
        {
            Statistics stat = new Statistics();
            Labels = new List<string> { };
            Data = new List<double> { };

            IEnumerable<Deposit> allDeposites = db.Deposites
                .Include(t => t.DepositInfo)
                .Include(p => p.ClientInfo);

            IEnumerable<DepositInfo> depositInfos = db.DepositInfos;
            foreach (DepositInfo d in depositInfos)
            {
                if (Labels.IndexOf(d.Name) == -1)
                {
                    Labels.Add(d.Name);
                }
            }
            int[] counters = new int[Labels.Count];
            foreach (Deposit d in allDeposites)
            {
                for (int i = 0; i < Labels.Count; i++)
                {
                    if (Labels[i] == d.DepositInfo.Name)
                    {
                        counters[i]++;
                    }
                }
            }
            int totalCount = allDeposites.Count();
            for(int i=0; i< Labels.Count; i++)
            {
                Data.Add(Math.Round((double)counters[i] / totalCount,3));
            }
        }
    }
}

```

```

    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace depositProject.Models
{
    public class User
    {
        public int Id { get; set; }
        public string Login { get; set; }
        public string Password { get; set; }
        public string Type { get; set; }
    }
}

using System.ComponentModel.DataAnnotations.Schema;
using System.Data.Entity;

namespace depositProject.Models
{
    public class UsersDB : DbContext
    {
        public UsersDB() : base("OracleDbContext") { }
        public DbSet<User> Users { get; set; }
        public DbSet<ClientInfo> ClientsInfos { get; set; }
        public DbSet<Deposite> Deposites { get; set; }
        public DbSet<DepositeInfo> DepositeInfoes { get; set; }
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.HasDefaultSchema("DEPOSITE");
            modelBuilder.Properties()
                .Where(p => p.PropertyType == typeof(string) &&
                    p.GetCustomAttributes(typeof(System.ComponentModel.DataAnnotations.MaxLengthAttribute), false).Length ==
                    0)
                .Configure(p => p.HasMaxLength(2000));
            Database.SetInitializer<UsersDB>(new DropCreateDatabaseIfModelChanges<UsersDB>());
            modelBuilder.Entity<ClientInfo>()
                .HasOptional(s => s.Deposite);

            modelBuilder.Entity<DepositeInfo>()
                .HasOptional(s => s.Deposite);

            modelBuilder.Entity<Deposite>().HasRequired(t => t.ClientInfo)
                .WithMany()
                .HasForeignKey(t => t.ClientInfoId)
                .WillCascadeOnDelete(false);

            modelBuilder.Entity<Deposite>().HasRequired(t => t.DepositeInfo)
                .WithMany()
                .HasForeignKey(t => t.DepositeInfoId)
                .WillCascadeOnDelete(false);
        }
    }
}

```

```
(function () {
    'use strict';
    var app = angular.module('DepositApp', ['ngRoute', 'authorizationModule', 'chart.js']);
    var config = function ($routeProvider) {
        $routeProvider
            .when("/login",
                { templateUrl: "/DepositApp/Views/login.html", controller: "loginController" })
            .when("/clientDeclaration",
                { templateUrl: "/DepositApp/Views/clientDeclaration.html", controller:
"clientDeclarationController" })
            .when("/acceptDepositInformation",
                { templateUrl: "/DepositApp/Views/acceptDepositInformation.html", controller:
"acceptDepositInformationController" })
            .when("/mainPage",
                { templateUrl: "/DepositApp/Views/mainPage.html", controller: "mainPageController" })
            .otherwise(
                { redirectTo: "/mainPage", controller: "mainPageController" })
            .otherwise(
                { redirectTo: "/login", controller: "loginController" });
    };

    app.config(config);
})();
(function (app) {
    app.directive('frontOfficeFrame', function () {
        return {
            restrict: 'E',
            scope: false,
            templateUrl: 'clientDeclaration.html',
            controller: function ($scope, clientOperationService, depositInfoesService, depositDataService,
$userProvider) {
                function ConvertUTCTimeToLocalTime(UTCDateString) {
                    var convertdLocalTime = new Date(UTCDateString);
                    var hourOffset = convertdLocalTime.getTimezoneOffset() / 60;
                    convertdLocalTime.setHours(convertdLocalTime.getHours() - hourOffset);
                    return convertdLocalTime;
                }

                depositDataService.getAcceptedDeposites()
                    .then(function successCallback(response) {
                        $scope.deposits = response.data;

                        if ($scope.deposits.length !== 0) {
                            $scope.message = "Наявні підтверджені депозити";
                            $scope.showMessage = false;
                            $scope.showTable = true;
                        } else {
                            $scope.message = "Відсутні підтверджені депозити";
                            $scope.showMessage = true;
                            $scope.showTable = false;
                        }
                    }, function errorCallback(response) { });
                depositDataService.getCreatedTodayDeposits(new Date())
                    .then(function successCallback(response) {
                        $scope.createdDeposits = response.data;

                        if ($scope.createdDeposits.length !== 0) {
                            $scope.message1 = "Сьогодні вже створювались депозити";
                        }
                    });
            }
        };
    });
});
```

```

        console.log($scope.createdDeposites);
        //$scope.showMessage = false;
        $scope.showCreatedTable = true;
    } else {
        $scope.message1 = "Сьогодні ще не створювалися депозити";
        //$scope.showMessage = true;
        $scope.showCreatedTable = false;
    }
}, function errorCallback(response) { });

$scope.reloadData = function () {
    depositDataService.getAcceptedDeposites()
        .then(function successCallback(response) {
            $scope.deposits = response.data;
            if ($scope.deposits.length !== 0) {
                $scope.message = "Наявні непідтверджені депозити";
                $scope.showMessage = false;
                $scope.showTable = true;
            } else {
                $scope.message = "Відсутні непідтверджені депозити";
                $scope.showMessage = true;
                $scope.showTable = false;
            }
        }, function errorCallback(response) { });
}

$scope.reloadCTDepositesData = function () {
    depositDataService.getCreatedTodayDeposites(new Date())
        .then(function successCallback(response) {
            $scope.createdDeposites = response.data;
            if ($scope.createdDeposites.length !== 0) {
                $scope.message1 = "Сьогодні вже створювалися депозити";
                //$scope.showMessage = false;
                $scope.showCreatedTable = true;
            } else {
                $scope.message1 = "Сьогодні ще не створювалися депозити";
                //$scope.showMessage = true;
                $scope.showCreatedTable = false;
            }
        }, function errorCallback(response) { });
}

$scope.showCreationOfDeposit = false;
$scope.showEditionOfDeposit = false;

$scope.openCreationOfDeposit = function () {
    $scope.showInputForm = false;
    $scope.messageStatus = false;
    $scope.showPrintForm = false;

    $scope.showCreationOfDeposit = true;
}

$scope.generatePDF = function (id) {
    depositDataService.getGeneratedContract(id)
        .then(function successCallback(response) {
            }, function errorCallback(response) { });
}

$scope.openEditionOfDeposit = function (deposit) {
    $scope.showEditionOfDeposit = true;

```

```

        $scope.deposit = deposit;
        $scope.deposit.StartDepositDate =
ConvertUTCTimeToLocalTime($scope.deposit.StartDepositDate);
        $scope.deposit.EndDepositDate =
ConvertUTCTimeToLocalTime($scope.deposit.EndDepositDate);

    }
    $scope.showCreationMessage = false;
    $scope.openCreationMessage = function () {
        $scope.showCreationMessage = true;
    }
}
});
(angular.module("DepositApp"));
(function (app) {
    app.directive('creationOfDeposit', function () {
        return {
            restrict: 'E',
            scope: false,
            templateUrl: 'createD.html',
            controller: function ($scope, $location,
depositDataService,clientOperationService,depositInfoesService) {
                $scope.clientInBase = false;
                function ConvertUTCTimeToLocalTime(UTCDateString) {
                    var convertdLocalTime = new Date(UTCDateString);
                    var hourOffset = convertdLocalTime.getTimezoneOffset() / 60;
                    convertdLocalTime.setHours(convertdLocalTime.getHours() - hourOffset);
                    return convertdLocalTime;
                }
                $scope.deposit = {};
                $scope.DepositInfo = {};
                $scope.showClientSearchForm = true;
                depositInfoesService.getInfoes()
                    .then(function successCallback(response) {
                        $scope.depositInfoes = response.data;
                    }, function errorCallback(response) {
                    });

                $scope.checkClientType = function (identificationCode, isValid) {
                    $scope.deposit = {};
                    if (isValid) {

                        $scope.messageStatus = false;
                        $scope.showInputForm = false;
                        $scope.cientInBase = false;
                        clientOperationService.checkClient(identificationCode)
                            .then(function successCallback(response) {
                                $scope.deposit.ClientInfo = response.data;
                                console.log($scope.deposit.ClientInfo);
                                $scope.message = "Клієнта знайдено";
                                $scope.messageStatus = true;
                                $scope.showInputForm = true;
                                $scope.clientInBase = true;
                                console.log($scope.clientInBase);
                            }, function errorCallback(response) {
                                $scope.deposit.ClientInfo = {};
                                $scope.deposit.ClientInfo.IndentificationCode = identificationCode;
                                console.log($scope.deposit.ClientInfo);

```

```

        $scope.message = "Клієнта не знайдено";
        $scope.messageStatus = true;
        $scope.showInputForm = true;
        $scope.clientInBase = false;
    });
}

}

$scope.submitDepositForm = function () {
    $scope.myForm.file.$setValidity("size", false);
}

$scope.submitDeposit = function (isValid, deposit) {

    console.log(isValid);
    if (isValid) {
        deposit.CreationDate = new Date();
        deposit.AcceptionDate = new Date(1754, 0, 1)
        deposit.Status = "created";
        if ($scope.clientInBase == true) {

            deposit.ClientInfoId = deposit.ClientInfo.Id;
            deposit.ClientInfo = null;

        }
        deposit.DepositInfoId = deposit.DepositInfo.Id;
        deposit.DepositInfo = null;
        depositDataService.saveDeposit(deposit)
            .then(function successCallback(response) {
                $scope.deposit = response.data;

                $scope.deposit.StartDepositDate =
                ConvertUTCTimeToLocalTime($scope.deposit.StartDepositDate);
                $scope.deposit.EndDepositDate =
                ConvertUTCTimeToLocalTime($scope.deposit.EndDepositDate);
                $scope.deposit.CreationDate =
                ConvertUTCTimeToLocalTime($scope.deposit.CreationDate);
                $scope.messageStatusCreation = "Депозит збережено";
                $scope.showPrintForm = true;
                $scope.showClientSearchForm = false;
                $scope.showInputForm = false;

                $scope.reloadData();
                $scope.reloadCTDepositsData();
                console.log($scope.deposit);

            }, function errorCallback() {
                $scope.messageStatusCreation = "Помилка запису";
            });
    }
}

$scope.todayDate = new Date();
console.log($scope.todayDate);
$scope.closeCreationOfDeposit = function () {
    $scope.showCreationOfDeposit = false;
    $scope.showClientSearchForm = true;
}

$scope.showCreationMessageInPopup = false;

```

```

        $scope.openCreationMessageInPopup = function () {
            $scope.showCreationMessageInPopup = true;
        }
    }
});
})(angular.module("DepositApp"));
(function (app) {
    app.directive('popUpMsg', function () {
        return {
            restrict: 'E',
            scope: false,
            templateUrl: 'AcceptD.html',
            controller: function ($scope, $location, depositDataService) {
                $scope.changeStatus = function (deposit) {
                    deposit.AcceptanceDate = new Date();
                    deposit.ClientInfoId = deposit.ClientInfo.Id;
                    deposit.DepositInfoId = deposit.DepositInfo.Id;
                    deposit.DepositInfo = null;
                    deposit.ClientInfo = null;
                    deposit.Status = "accepted";
                    console.log(deposit);
                    depositDataService.updateDeposit(deposit)
                        .then(function successCallback(response) {
                            $scope.message = "Дані оновлено";
                            $scope.closePopUp();
                            $scope.reloadData();
                            $scope.reloadCTDepositsData();
                            $scope.reloadNotAcceptedDeposits();
                        }, function errorCallback() {
                            $scope.message = "Помилка запису";
                        });
                }
                $scope.rejectDeposit = function (deposit) {
                    deposit.ClientInfoId = deposit.ClientInfo.Id;
                    deposit.DepositInfoId = deposit.DepositInfo.Id;
                    deposit.DepositInfo = null;
                    deposit.ClientInfo = null;
                    deposit.Status = "notAccepted";
                    console.log(deposit);
                    depositDataService.updateDeposit(deposit)
                        .then(function successCallback(response) {
                            $scope.message = "Дані оновлено";
                            $scope.closePopUp();
                            $scope.reloadData();
                            $scope.reloadCTDepositsData();
                            $scope.reloadNotAcceptedDeposits();
                        }, function errorCallback() {
                            $scope.message = "Помилка запису";
                        });
                }
                $scope.closePopUp = function () {
                    $scope.showPopUpMsg = false;
                }
            }
        };
    });
})(angular.module("DepositApp"));
(function (app) {
    app.directive('empty', function () {

```



```

return {
  require: 'ngModel',
  restrict: "",
  link: function (scope, elm, attrs, ctrl) {
    ctrl.$validators.empty = function (modelValue, viewValue) {
      if (ctrl.$isEmpty(modelValue)) {
        return false;
      }
      return true;
    };
  }
};
});
})(angular.module("DepositApp"));
```

					КПІ ІС-5124.1181-с.ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис)      Т.О. Тєлишева  
(ініціали, прізвище)

“16” квітня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_  
(підпис)      О.А. Павлов  
(ініціали, прізвище)

“17” квітня 2019 р.

Підсистема підтримки діяльності малого та середнього бізнесу у  
складі автоматизованої банківської системи “БАРС”

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП ІС-5124.1181-с.ТЗ*

на 12 сторінках

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>3</b>
1.1	Повне найменування системи та її умовне позначення .....	3
1.2	Найменування організації-замовника та організацій-учасників робіт .....	3
1.3	Перелік документів, на підставі яких створюється підсистема .....	3
1.4	Планові терміни початку і закінчення роботи зі створення підсистеми .....	4
<b>2</b>	<b>ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ ПІДСИСТЕМИ.....</b>	<b>5</b>
2.1	Призначення підсистеми .....	5
<b>3</b>	<b>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ .....</b>	<b>6</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>7</b>
4.1	Вимоги до функціональних характеристик.....	7
4.2	Вимоги до надійності .....	8
4.3	Вимоги до складу і параметрів технічних засобів.....	8
<b>5</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ.....</b>	<b>10</b>
<b>6</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ ПІДСИСТЕМИ .....</b>	<b>11</b>
6.1	Види випробувань.....	11

					ДП ІС-5124.1181-с.ТЗ					
Зм.	Арк.	Прізвище	Підпис	Дата						
Розроб.		Сборик А.Ю.			Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”		Літ.	Арк.	Аркушів	
Перевірів		Телишева Т.О							2	12
							НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ кафедра АСОІУ гр. ІС-51			
Н. кон.		Москаленко Н.В.								
Затв.		Павлов О.А.								

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Повне найменування системи та її умовне позначення

**Повна назва підсистеми:** Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”.

**Умовне позначення:** Депозитний модуль.

## 1.2 Найменування організації-замовника та організацій-учасників робіт

Генеральними замовниками проекту являється кафедра Автоматизованих систем обробки інформації та управління НТУУ “КПІ” та ТОВ “Уніті-Барс”.

Розробником підсистеми є студент групи ІС-51 факультету інформатики та обчислювальної техніки кафедри АСОІУ НТУУ «КПІ» імені Ігоря Сікорського Сборик Антон Юрійович.

## 1.3 Перелік документів, на підставі яких створюється підсистема

Під час розробки підсистеми і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Планові терміни початку і закінчення роботи зі створення підсистеми

Плановий термін початку роботи над створенням підсистеми підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС” – 15 квітня 2019 року.

Плановий термін по закінченню роботи над створенням підсистеми підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС” – не пізніше 3 червня 2019 року.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ ПІДСИСТЕМИ

### 2.1 Призначення підсистеми

Призначенням підсистеми є підтримка малого та середнього бізнесу, аналіз популярності наявних депозитних продуктів, шляхом перегляду побудованих діаграм, розрахунку кінцевої суми, отриманої від депозитного вкладу та знаходження дати повернення депозиту.

#### Цілі та задачі підсистеми

Основними цілями розробки є:

- зменшення часу на відкриття депозитів представникам малого бізнесу, середнього бізнесу та працівниками банку за рахунок автоматизації процесу формування документів на основі введених даних;
- пониження ймовірності помилок при оформленні документів працівниками банку за рахунок валідації форм введення даних та додаткової перевірки даних збережених до бази даних працівниками бек офісу.

Для досягнення поставлених цілей мають бути вирішені такі задачі:

- створення функціоналу вибору для клієнта типу депозиту зі списку наявних депозитних продуктів;
- розробка функціоналу аналізу вибору наявних депозитних продуктів для працівників бек офісу банку, візуалізація аналітики
- розробка бази даних для клієнтів, відкритих депозитів;
- створення функціоналу для пошуку клієнта в базі даних банку при оформленні депозита;
- розробка алгоритмів знаходження дати повернення депозиту, розрахунку прибутку отриманого від депозитного вкладу.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації у дипломній роботі є процес оформлення депозита. Автоматизуватися повинен процес оформлення депозиту з подальшим підтвердженням та друком договору та заяви клієнта на основі введених даних під час відкриття депозиту.

В даний час існує декілька подібних підсистем в автоматизованих банківських системах Грант, ИСАОД та ін[3-6]. Основною проблемою використання існуючих рішень є відсутність перегляду статистики по існуючим депозитним продуктам, а також не повнота функціоналу та застаріли технології, що використовуються в деяких з них.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Підсистема забезпечує наступний функціонал:

- відкриття депозиту;
- редагування депозиту;
- підтвердження депозиту;
- друк документів;
- розрахунок кінцевої суми нарахованої по відсотковій ставці;
- знаходження дати повернення вкладу;
- розрахунок та візуалізація частотності вибору типів депозитних продуктів.

Підсистема повинна надавати можливість відкриття депозиту працівнику фронт офісу банку. Форма введення даних про клієнта повинна відображатись у діалоговому вікні. Під час початку введення даних про депозит на формі введення даних про депозит повинне відображатись поле пошуку клієнта в базі даних банку. Після введення даних і відкриття депозиту система повинна зберегти дані про клієнта та депозит.

Підсистема повинна надати можливість редагування створених депозит працівнику фронт офісу банку. Форма редагування даних повинна відображатись в діалоговому вікні та містити кнопку підтвердження редагування та кнопку скасування редагування. Після натискання кнопки збереження редагування система повинна оновлювати дані про депозит.

Підсистема повинна надати можливість друку документів на основі введених даних під час відкриття депозиту працівнику фронт офісу банку. Документи повинні формуватися на основі даних, збережених в базу даних, та повинні завантажуватися при натисканні відповідних кнопок друку.

Підсистема повинна надати можливість працівнику бек офісу підтверджувати відкриті депозити. Форма підтвердження повинна

					ДП ІС-5124.1181-с.ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



відкриватись в діалоговому вікні та повинна містити поле коментаря, що може залишити оператор бек офісу. Після підтвердження або скачування депозиту, повинен змінитись статус депозиту та зберегтись до база даних.

Підсистема повинна надати оператору фронт офісу можливість розраховувати кінцеву суму нарахованої по відсотковій ставці та дату повернення депозиту оператору фронт офісу. Кінцева сума і дата повернення повинна розраховуватись під час формування депозитного договору. Після натискання кнопки друку, автоматично шукані величини мають бути заповненні у депозитному договорі.

Підсистема повинна надати можливість оператору бек офісу розрахувати популярність існуючих депозитних продуктів. Пошук частотності вибору типів депозитних рахунків повинно здійснюватися після натискання кнопки “Побудувати діаграму”. На головній сторінці оператора бек офісу повинна будуватися кругова діаграма та при наведенні курсора на деяку з ділянок повинна відображатися частотність та назва відповідного депозитного договору.

#### 4.2 Вимоги до надійності

Підсистема повинна не конфліктувати з іншими підсистемами автоматизованої банківської системи, не знищувати та не спотворювати дані в базі даних автоматизованої банківської системи. Підсистема повинна відпрацьовувати всі дії, які вказані в функціональних вимогах, не викликаючи помилок та не завершуючи аварійно свою роботу. Підсистема повинна мати зручний інтерфейс.

#### 4.3 Вимоги до складу і параметрів технічних засобів

Для коректного функціонування підсистеми було виділено наступні вимоги до технічного забезпечення

1. Сервер на якому буде встановлено веб застосування та база даних:

					ДП ІС-5124.1181-с.ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- процесор 4 ядра з тактовою частотою більше 2.5 ГГц;
- швидкість інтернету 1 Гб;
- оперативна пам'ять 8 Гб;
- об'єм дискового простору 1 Тб.

Додатково має бути встановлене таке програмне забезпечення:

- 64 розрядна операційна система Windows 10;
- база даних Oracle Database 12c Express Edition;
- .Net Framework 4.5 і вище.

## 2. Робочий комп'ютер працівника банку:

- процесор з тактовою частотою не нижче 2 ГГц;
- достатній об'єм оперативної пам'яті (не менше 4 Гб);
- підключення до мережі Інтернет;
- Підтримка мови JavaScript в браузері;
- інші складові можуть мати будь-які параметри, тому що вони не значним чином впливають на роботу програми.

Обов'язкова комп'ютерна периферія, до складу якої входить:

- монітор;
- мишка;
- клавіатура;
- принтер.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки підсистеми підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”.

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	01.04.2019	ТЗ підготовлено
2.	Розробка сценарію роботи	10.04.2019	Сценарій розроблено
3.	Технічне проектування – функціональність, розробка концепції, задачі, цілі тощо	15.04.2019	Функціональність, модулі, задачі, цілі визначено
4.	Узгодження з керівником інтерфейсу користувача	20.04.2019	Інтерфейс узгоджено
5.	Розробка інформаційного забезпечення	25.04.2019	Інформаційне забезпечення розроблено
6.	Розробка програмного забезпечення	10.05.2019	Програмне забезпечення розроблено
7.	Налагодження програми	13.05.2019	Налагоджена програма
8.	Тестування програми	19.05.2019	Програма протестована
9.	Здача готового програмного продукту замовнику	03.06.2019	Готове ПЗ

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ ПІДСИСТЕМИ

### 6.1 Види випробувань

Правильність роботи підсистеми буде перевірятися за допомогою функціонального тестування. При тестуванні буде перевірено основні функції підсистеми та роботу підсистеми в цілому.

- відкриття депозиту;
- редагування депозиту;
- підтвердження депозиту;
- друк документів;
- розрахунок кінцевої суми нарахованої по відсотковій ставці;
- знаходження дати повернення вкладу;
- розрахунок та візуалізація частотності вибору типів депозитних продуктів.

Очікуваним результатом тестування відкриття депозиту є успішне збереження даних про депозит до бази даних,

Очікуваним результатом тестування редагування депозиту є успішне збереження оновлених даних про депозит до бази даних.

Очікуваним результатом тестування підтвердження депозиту є успішне збереження коментаря оператора бек офісу та зміни статусу депозиту на “підтверджений” в базі даних, поява щойно підтвердженого депозиту в списку підтверджених депозитів на сторінці фронт офісу.

Очікуваним результатом тестування друку документів є успішне формування та збереження документів до комп’ютера користувача.

Очікуваним результатом тестування розрахунку кінцевої суми нарахованої по відсотковій ставці є успішне розрахування кінцевої суми та її наявність в сформованому депозитному договорі.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Очікуваним результатом тестування знаходження дати повернення вкладу є успішне знаходження дати та її наявність в сформованому депозитному договорі.

Очікуваним результатом тестування розрахунку та візуалізації частотності вибору типів депозитних продуктів є успішне розрахування частотності вибору типів депозитних продуктів та відображення діаграми на сторінці бек офісу.

					ДП ІС-5124.1181-с.ТЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис)      Т.О. Тєлишева  
(ініціали, прізвище)

“ 13 ” травня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_  
(підпис)      О.А. Павлов  
(ініціали, прізвище)

“ 14 ” травня 2019 р.

Підсистема підтримки діяльності малого та середнього бізнесу у  
складі автоматизованої банківської системи “БАРС”

**ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

Шифр ДП ІС-5124.1181-с.ПМВ

на 17 сторінках

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ОБ'ЄКТ ВИПРОБУВАННЯ.....</b>	<b>3</b>
1.1	Найменування програми .....	3
1.2	Область застосування.....	3
1.3	Умовне позначення програми .....	3
<b>2</b>	<b>МЕТА ВИПРОБУВАНЬ .....</b>	<b>4</b>
<b>3</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ .....</b>	<b>5</b>
3.1	Вимоги до функціональних характеристик .....	5
3.1.1	Вимоги до складу виконуваних функцій .....	5
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....</b>	<b>6</b>
<b>5</b>	<b>СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ.....</b>	<b>7</b>
<b>6</b>	<b>МЕТОДИ ВИПРОБУВАНЬ.....</b>	<b>17</b>

					<b>ДП ІС-5124.1181-с.ПМВ</b>			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Сборик А.Ю.			Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи "БАРС"	Літ.	Арк.	Аркушів
Перевірів.		Тэлишева Т.О.					2	17
Н. кон.		Москаленко Н.В.				НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ кафедра АСОІУ гр. ІС-51		
Затв.		Павлов О.А.						

## 1 ОБ'ЄКТ ВИПРОБУВАННЯ

### 1.1 Найменування програми

Повне найменування – Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”.

### 1.2 Область застосування

Призначенням підсистеми є підтримка малого та середнього бізнесу, аналіз популярності наявних депозитних продуктів, шляхом перегляду побудованих діаграм, розрахунку кінцевої суми, отриманої від депозитного вкладу та знаходження дати повернення депозиту.

### 1.3 Умовне позначення програми

Умовне позначення програми – Депозитний модуль.

					ДП ІС-5124.1181-с.ПМВ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		



## 2 МЕТА ВИПРОБУВАНЬ

Метою випробувань являється перевірка відповідності функцій підсистеми відкриття депозитів для малого та середнього бізнесу вимогам технічного завдання.

					ДП ІС-5124.1181-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Вимоги до функціональних характеристик

Підсистема має забезпечувати наступний функціонал:

- відкриття депозиту;
- редагування депозиту;
- підтвердження депозиту;
- друк документів;
- розрахунок кінцевої суми нарахованої по відсотковій ставці;
- знаходження дати повернення вкладу;
- розрахунок та візуалізація частотності вибору типів депозитних

продуктів.

##### 3.1.1 Вимоги до складу виконуваних функцій

Підсистема має задовольняти наступним вимогам:

- не конфліктувати з іншими підсистемами автоматизованої банківської системи;
- не знищувати та не спотворювати дані в базі даних автоматизованої банківської системи;
- не завершувати аварійно свою роботу;
- зручний інтерфейс.

					ДП ІС-5124.1181-с.ПМВ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації повинні входити наступні документи:

- титульний аркуш;
- лист завдання;
- календарний план;
- відомість дипломного проекту;
- технічне завдання;
- пояснювальна записка;
- титульний аркуш пояснювальної записки;
- анотація українською мовою;
- анотація англійською мовою;
- зміст;
- вступ;
- загальні положення;
- інформаційне забезпечення;
- математичне забезпечення;
- програмне та технічне забезпечення;
- висновки;
- список використаної літератури;
- програма та методика випробувань;
- список скорочень;
- графічний матеріал.

Усі документи повинні бути оформлені за відповідними стандартами.

					ДП ІС-5124.1181-с.ПМВ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

В процесі тестування буде перевірена уся функціональність підсистеми. У наступних таблицях наведений перелік випробувань основного функціоналу (табл. 1.1 – 1.15).

Таблиця 1.1 – Перевірка відкриття форми відкриття депозиту

Мета тесту:	Перевірка функції «Відкриття форми відкриття депозиту»
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Відкрити новий депозит”
Очікуваний результат:	Відкрилась форма для створення нового депозиту у діалоговому вікні
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу” з активною формою для створення нового депозиту у діалоговому вікні

Таблиця 1.2 – Перевірка пошуку клієнта в базі даних банку

Мета тесту:	Перевірка функції «Пошук клієнта в базі даних банку»
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	Код ідентифікації клієнта

Продовження таблиці 1.2

Схема проведення тесту:	Натиснути кнопку “Пошук клієнта”
Очікуваний результат:	В формі у разі наявності клієнта у відповідних полях відображається інформація клієнта, а у разі відсутності поля пусті та код ідентифікації клієнта переноситься у наступне поле.
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні

Таблиця 1.3 – Перевірка відкриття депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Відкриття депозиту»</b>
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Відкрити депозит”
Очікуваний результат:	Поля введення даних зникли, з’явилися 2 кнопки:”Друк анкети користувача”,”Друк заяви на відкриття депозиту”
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні

Таблиця 1.4 – Перевірка переходу до сторінки роботи з модулем користувача типу “Оператор фронт офісу” з форми відкриття депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Перехід до сторінки роботи з модулем користувача типу “Оператор фронт офісу” з форми відкриття депозиту»</b>
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Закрити”
Очікуваний результат:	Форма для створення нового депозиту зникла, відкрилась сторінка роботи з модулем користувача типу “Оператор фронт офісу”, додано запис в таблицю депозитів за сьогодні зі статусом “створений”
Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор фронт офісу” з активною формою для створення нового депозиту у діалоговому вікні

Таблиця 1.5 – Перевірка валідності форми створення нового депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Перевірка валідності форми створення нового депозиту»</b>
Початковий стан підсистеми	Відкрита форма створення нового депозиту у діалоговому вікні

Продовження таблиці 1.5

Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Відкрити депозит”
Очікуваний результат:	З’явилися відповідні помилки про хибність або відсутність введених даних
Стан підсистеми після проведення випробувань:	Відкрита форма створення нового депозиту у діалоговому вікні з вказаними помилками вводу

Таблиця 1.6 – Перевірка редагування даних депозиту

<b>Мета тесту:</b>	<b>Перевірка функції «Редагування даних депозиту»</b>
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”
Вхідні данні:	Id обраного депозиту
Схема проведення тесту:	Натиснути кнопку “Редагувати”
Очікуваний результат:	Відкрилась форма редагування обраного депозиту
Стан підсистеми після проведення випробувань:	Відкрита форма редагування депозиту у діалоговому вікні

Таблиця 1.7 – Перевірка редагування депозиту

Мета тесту:	Перевірка функції «Редагування депозиту»
Початковий стан підсистеми	Відкрита форма редагування депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Підтвердити редагування”
Очікуваний результат:	Перехід до сторінки роботи з модулем користувача типу “Оператор фронт офісу” та збереження змін
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу “Оператор фронт офісу”

Таблиця 1.8 – Перевірка валідності форми редагування депозиту

Мета тесту:	Перевірка функції «Перевірка валідності форми редагування депозиту»
Початковий стан підсистеми	Відкрита форма редагування депозиту у діалоговому вікні
Вхідні данні:	Дані про клієнта, тип депозиту, депозит
Схема проведення тесту:	Натиснути кнопку “Підтвердити редагування”



Продовження таблиці 1.8

Очікуваний результат:	З'явилися відповідні помилки про хибність або відсутність введених даних
Стан підсистеми після проведення випробувань:	Відкрита форма редагування депозиту у діалоговому вікні

Таблиця 1.9 – Перевірка друку договору

Мета тесту:	Перевірка функції «Друк договору»
Початковий стан підсистеми	Відкрита сторінка роботи з модулем користувача типу «Оператор фронт офісу»
Вхідні данні:	Id обраного депозиту
Схема проведення тесту:	Натиснути кнопку «Друк»
Очікуваний результат:	Відкрилось діалогове вікно з документом для друку
Стан підсистеми після проведення випробувань:	Відкрита сторінка роботи з модулем користувача типу «Оператор фронт офісу» з відкритим діалоговим вікном з документом для друку

Таблиця 1.10 – Перевірка відкриття форми підтвердження депозиту

Мета тесту:	Перевірка функції «Відкриття форми підтвердження депозиту»
Початковий стан підсистеми	Відкрита сторінка для роботи з модулем користувача типу «Оператор бек офісу»
Вхідні данні:	Id обраного депозиту

## Продовження таблиці 1.10

Схема проведення тесту:	Натиснути кнопку “Переглянути”
Очікуваний результат:	Відкрита форма підтвердження депозиту у діалоговому вікні
Стан підсистеми після проведення випробувань:	Відкрита форма підтвердження депозиту у діалоговому вікні

## Таблиця 1.11 – Перевірка підтвердження депозиту

Мета тесту:	Перевірка функції «Підтвердження депозиту»
Початковий стан підсистеми	Відкрита форма підтвердження депозиту у діалоговому вікні
Вхідні данні:	Коментар оператора бек-офісу
Схема проведення тесту:	Натиснути кнопку “Підтвердити”
Очікуваний результат:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”, дані про депозит збережено, додано запис в таблицю депозитів за сьогодні зі статусом “підтверджений”
Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”

Таблиця 1.12 – Перевірка відмови у підтвердженні

Мета тесту:	Перевірка функції «Відмова у підтвердженні»
Початковий стан підсистеми	Відкрита форма підтвердження депозиту у діалоговому вікні
Вхідні данні:	Коментар оператора бек-офісу
Схема проведення тесту:	Натиснути кнопку “Не підтверджувати”
Очікуваний результат:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”, дані про депозит збережено, додано запис в таблицю депозитів за сьогодні зі статусом “не підтверджений”
Стан підсистеми після проведення випробувань:	Відкрита сторінка для роботи з модулем користувача типу “Оператор бек офісу”

Таблиця 1.13 – Перевірка авторизації в системі

Мета тесту:	Перевірка функції «Авторизація в системі»
Початковий стан підсистеми	Відкрита сторінка авторизації
Вхідні данні:	Логін, пароль
Схема проведення тесту:	Натиснути кнопку “Авторизуватися”
Очікуваний результат:	Перехід на сторінку відповідно до ролі користувача

Продовження таблиці 1.13

Стан підсистеми після проведення випробувань:	Відкрита сторінка відповідно до ролі користувача
---	--

Таблиця 1.14 – Перевірка валідності форми авторизації

Мета тесту:	Перевірка функції «Перевірка валідності форми авторизації»
Початковий стан підсистеми	Відкрита сторінка авторизації
Вхідні данні:	Логін, пароль, що відсутні в БД, або пусті
Схема проведення тесту:	Натиснути кнопку “Авторизуватися”
Очікуваний результат:	Повідомлення про відповідну помилку входу
Стан підсистеми після проведення випробувань:	Відкрита сторінка авторизації

Таблиця 1.15 – Перевірка побудови діаграми частотності вибору типів депозиту

Мета тесту:	Перевірка функції «Побудова діаграми частотності вибору типів депозиту»
Початковий стан підсистеми	Відкрита сторінка депозитного модуля оператора бек офісу
Вхідні данні:	-
Схема проведення тесту:	Натиснути кнопку “Побудувати діаграму”
Очікуваний результат:	Побудована діаграма частотності вибору типів депозиту

## Продовження таблиці 1.15

Стан підсистеми після проведення випробувань:	Відкрита сторінка депозитного модуля оператора бек офісу
---	--

## 6 МЕТОДИ ВИПРОБУВАНЬ

Метод даних випробувань – ручне тестування по наведеним в таблицях 1.1-1.15 тестовим сценаріям.

Результати всіх випробувань еквівалентні до очікуваних результатів.

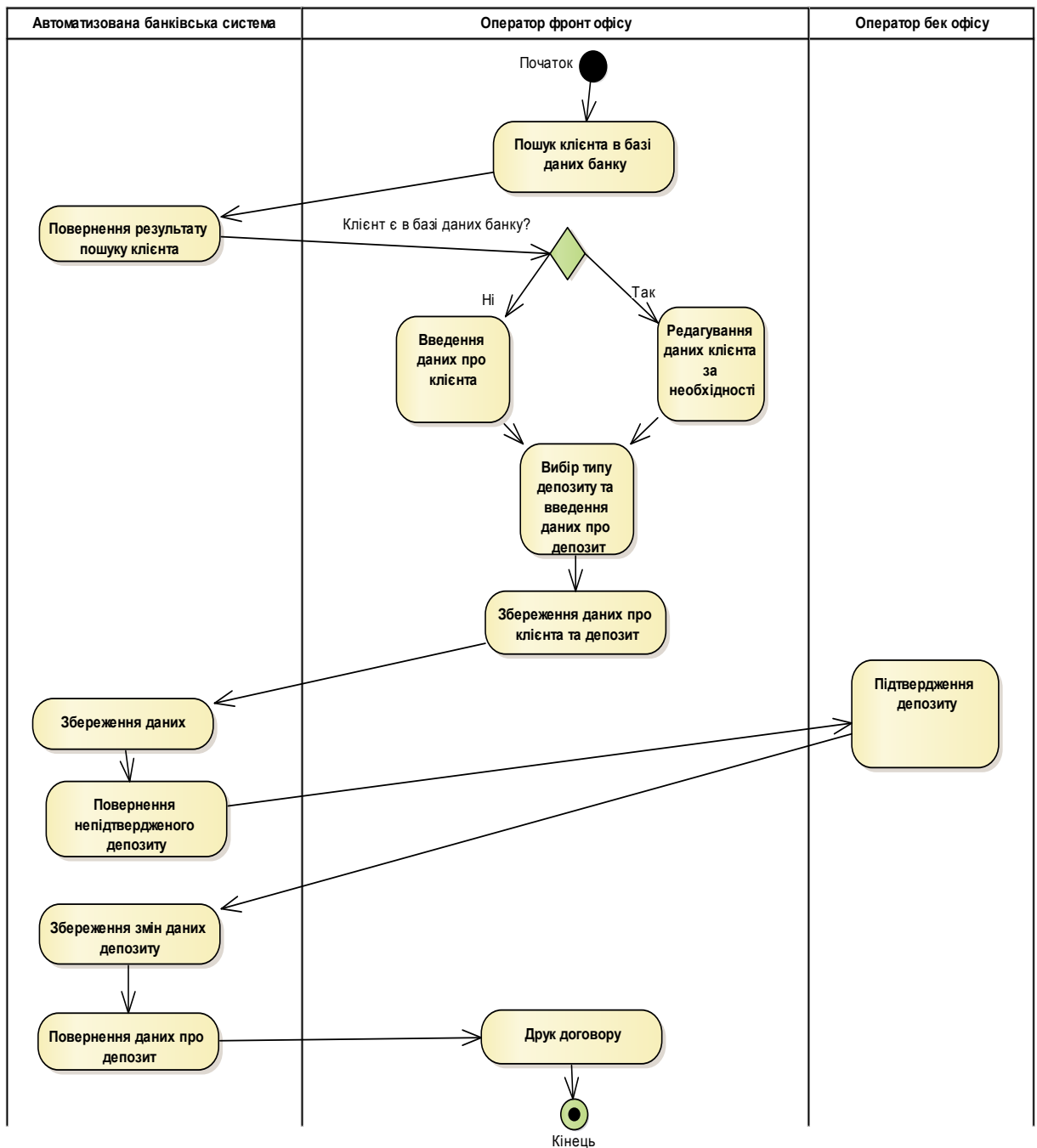
					ДП ІС-5124.1181-с.ПМВ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

# **Графічний матеріал до дипломного проекту**

на тему: *Підсистема підтримки діяльності малого та середнього  
бізнесу у складі автоматизованої банківської системи*  
*“БАРС”*

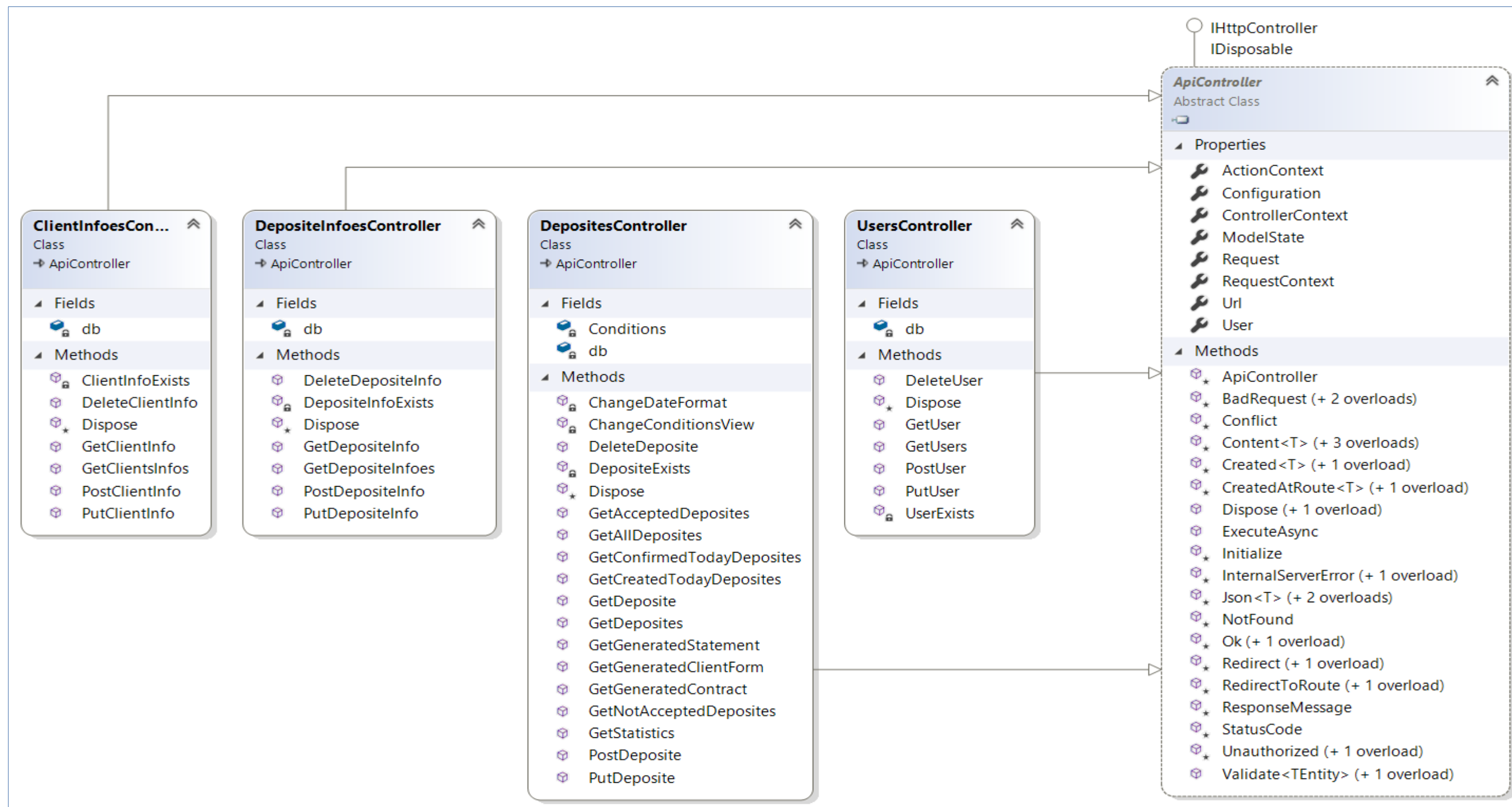
Київ – 2019 року

act activity

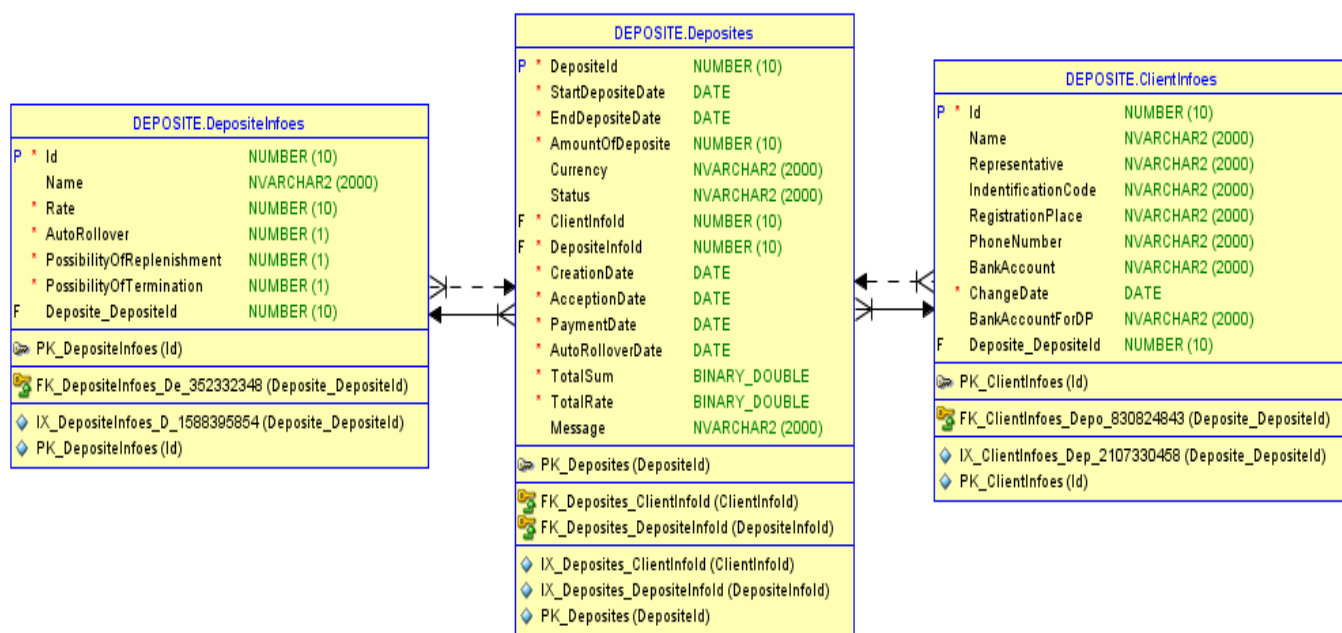


					ДП ІС-5124.1181-с.ССД						
					Схема структурна діяльності	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Сборик А.Ю										
Перевірів	Телишева Т.О.					Аркуш 1		Аркушів 1			
Т. кон.					Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51					
Н. кон.	Москаленко Н.В										
Затвердив	Телишева Т.О.										

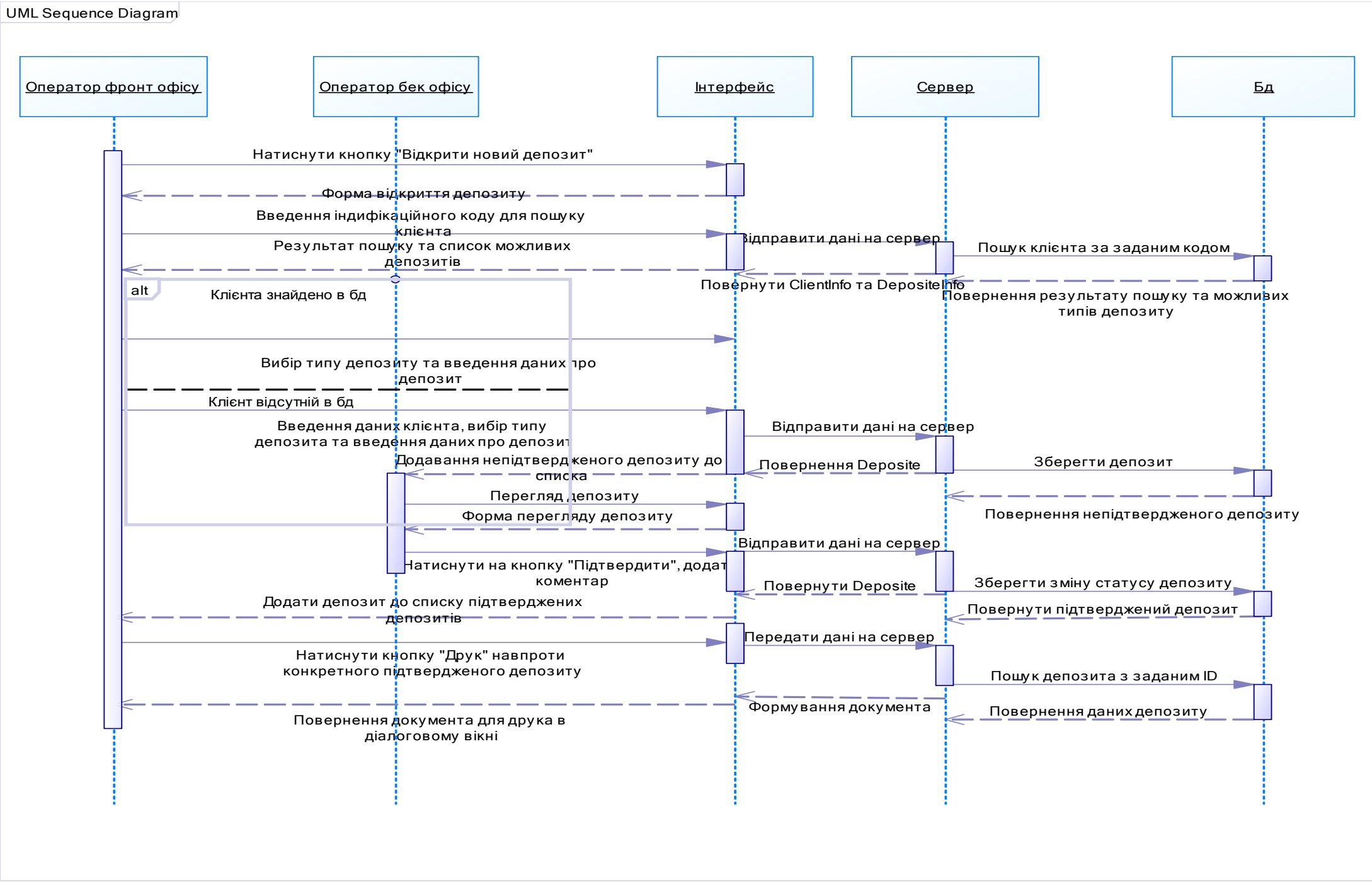




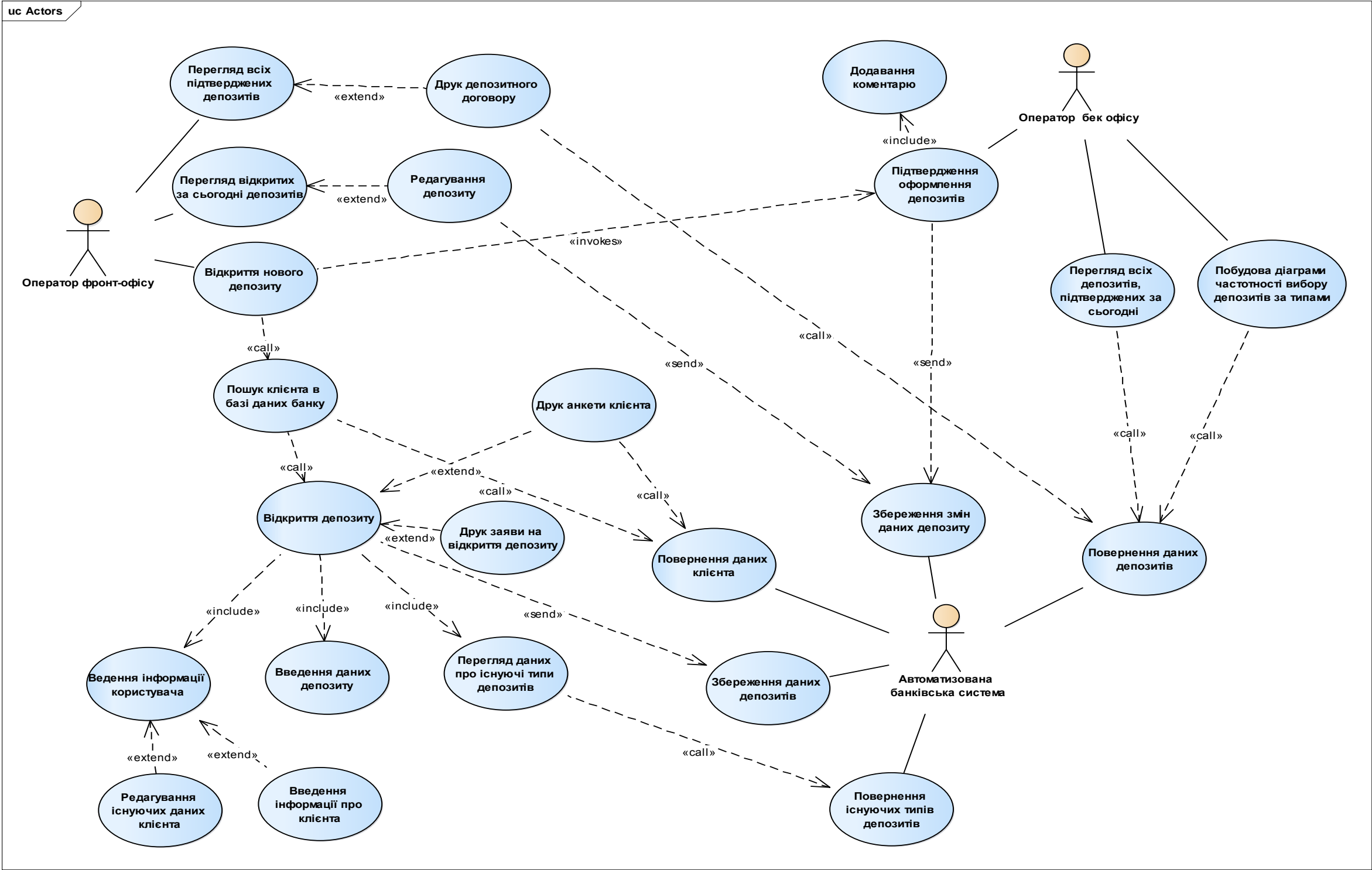
					ДП IC-5124.1181-с.ССК				
					Схема структурна класів програмного забезпечення	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив	Сборик А.Ю								
Перевірів	Телишева Т.О.								
Т. кон.						Аркуш 1		Аркушів 1	
					Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. IC-51			
Н. кон.	Москаленко Н.В.								
Затвердив	Телишева Т.О.								



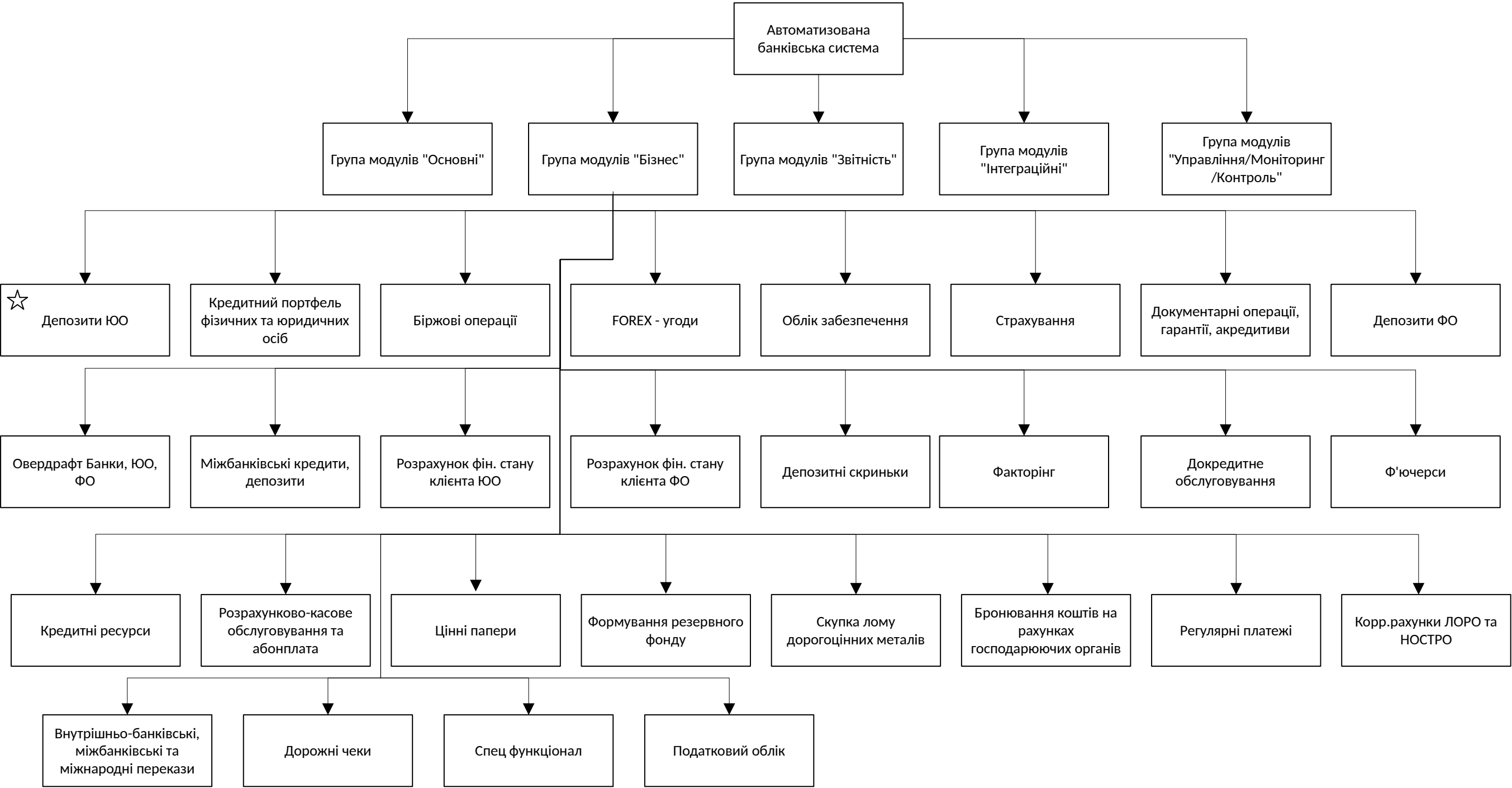
					ДП IC-5124.1181-с.СБД							
					Схема бази даних	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Сборик А.Ю											
Перевірив	Телишева Т.О.											
Т. кон.					Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”	Аркуш 1			Аркушів 1			
Н. кон.		Москаленко Н.В.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. IC-51						
Затвердив		Телишева Т.О.										



					ДП ІС-5124.1181-с.ССП			
					Схема структурна послідовності	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Сборик А.Ю.							
Перевірів	Телишева Т.О.							
Т. кон.						Аркуш 1		Аркушів 1
Н. кон.	Москаленко Н.В.				Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи "БАРС"	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Затвердив	Телишева Т.О.							



					ДП ІС-5124.1181-с.ССВ						
					Схема структурна варіантів використань	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Сборик А.Ю									
Перевірів		Телишева Т.О.									
Т. кон.											
					Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”	Аркуш 1			Аркушів 1		
Н. кон.		Москаленко Н.В				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51					
Затвердив		Телишева Т.О.									



					ДП ІС-5124.1181-с.САС								
					Схема структурна АС	Літера			Маса		Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив		Сборик А.Ю											
Перевірив		Телишева Т.О.											
Т. кон.													
						Аркуш 1			Аркушів 1				
Н. кон.		Москаленко Н.В.			Підсистема підтримки діяльності малого та середнього бізнесу у складі автоматизованої банківської системи “БАРС”					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Затвердив		Телишева Т.О.											